Ammar Alkassar
Melanie Volkamer (Eds.)

LNCS 4896

E-Voting and Identity

First International Conference, VOTE-ID 2007
Bochum, Germany, October 2007
Revised Selected Papers

Springer

Lecture Notes in Computer Science 4896

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Ammar Alkassar Melanie Volkamer (Eds.)

E-Voting and Identity

First International Conference, VOTE-ID 2007
Bochum, Germany, October 4-5, 2007
Revised Selected Papers

Volume Editors

Ammar Alkassar
Sirrix AG security technologies
Im Stadtwald D3.2, 66123 Saarbrücken, Germany
E-mail: a.alkassar@sirrix.com

Melanie Volkamer
University of Passau, Institute of IT-Security and Security Law
Innstr. 43, 94032 Passau, Germany
E-mail: volkamer@uni-passau.de

Library of Congress Control Number: 2007941815

CR Subject Classification (1998): E.3, D.4.6, C.2, J.1, H.2.0, K.5.2, K.6.5, K.4.4

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-77492-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-77492-1 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12210206 06/3180 5 4 3 2 1 0

Foreword

Voting and identity have a very delicate relationship. Only a few processes depend so much on an identity management respecting the fine line between reliable identification and reliable non-identifiability each at its part during the process. And only a few processes may change their outer appearance so much with the advent of new IT as voting and identity management do.

So it was no surprise in FIDIS, the interdisciplinary Network of Excellence working on the Future of Identity in the Information Society, when Ammar Alkassar proposed analyze the technical, socio-ethical and legal relations between Identity and E-Voting as part of Sirrix's activity in FIDIS.

There are many reasons for doing this, e.g., the open question of the implications of identity and identification to the emerging field of E-Government and E-Democracy, especially E-Voting. Issues to be discussed are from several domains, e.g., is identity fraud a crucial matter in E-Voting? What is the trade-off between anonymity and free speech vs. content-related offences? Is it appropriate to use ID cards or health-insurance cards with digital identities for citizen tasks or voting? What about using SIM cards? Can we employ biometrics for identification purposes with respect to E-Democracy?

Last but not least nearly all areas of E-Government rely on a reliable link between the citizens and their governments and administrations. However, in contrast to business processes, the effects are much more crucial: Identity fraud may cause more problems than in the business domain; the consequences of misuse cannot be measured just by financial means.

With these and many other issues at stake it was great to see VOTE-ID 2007 become such a great success with high-quality papers and discussions. It is a great pleasure to thank all the submitters, the Program Committee, and especially the Program Chairs Ammar Alkassar (Sirrix AG security technologies) and Melanie Volkamer (Institute of IT-Security and Security Law, University Passau) for the tremendous work in getting this conference off the ground.

November 2007

Kai Rannenberg
Goethe University Frankfurt
FIDIS Co-ordination

Preface

Electronic voting has been one of the most controversial topics of discussion in the IT security community for the past 20 years. During the 1980s, the discussion was characterized by the development of new, powerful cryptographic schemes and protocols. These were driven by the necessity to meet the requirements for replacing the former analog systems with newer election systems and e-voting technologies.

However, recurring problems with the election systems that were deployed, as well as inherent weaknesses, have burdened the argument for pushing forward. Now, after what could be characterized as a turbulent wave of pros and cons, the discussion focus has moved to address how the democratic spirit of elections can be respected in full, while also gaining the confidence of the public in the latest voting systems.

With respect to this new discussion, it was quite natural for the FIDIS Network of Excellence (NoE) to address the topic of E-Voting and Identity as well as its relevance in democratic society.

"Future of IDentity in the Information Society" (FIDIS) is a project funded by the European Commission. The network consists of 24 partners from 11 European countries collaborating on topics such as privacy, data protection, profiling and identity in both the public and private sectors.

An important aspect of the FIDIS NoE, as well as the recent conference, is to provide a highly-interdisciplinary forum for researchers stemming from various fields and organizations. Hence, the Program Committee was selected to represent leading experts in the related areas of cryptography, voting systems and ID management as well as legal and social sciences.

The conference was successful in bringing together researchers from universities and research institutes as well as practitioners from industry and electoral boards to discuss the central aspects of e-voting as well as the more pragmatic issues.

We would like to thank Berry Schoenmakers from the Technical University in Eindhoven (The Netherlands) for his excellent keynote on "E-Voting Crises" and also the panel members of the panel discussion: Klaus Brunnstein (University of Hamburg, Germany), Hans van Wijk (NEDAP, The Netherlands), Robert Stein (Head of Election Division, Federal Ministry of Interior, Austria) and Craig Burton (Everyone Counts).

We would like to extend a special thanks to Cline Fischer, who was kind enough to arrange the conference venue and take care of the administrative tasks which allowed the conference to run so smoothly. The conference was hosted by Sirrix AG and held at the European Center for IT-Security in Bochum.

November 2007

Ammar Alkassar
Melanie Volkamer

Organization

Program Chairs

Ammar Alkassar	Sirrix AG, Germany
Melanie Volkamer	Passau University, Germany

Program Committee

Josh Benaloh	Microsoft, USA
Rüdiger Grimm	Koblenz-Landau University, Germany
Marit Hansen	ICPP, Germany
Dirk Heckmann	Passau University, Germany
David-Olivier Jaquet-Chiffelle	University of Applied Sciences of Bern, Switzerland
Frank Koob	German Federal Office for Information Security, Germany
Robert Krimmer	evoting.cc, Austria
Ronald Leenes	Tilburg University, Netherlands
Helger Lipmaa	University College London, UK
Sjouke Mauw	Luxemburg University, Luxemburg
Margaret McGaley	NUI Maynooth, Ireland
Lilian Mitrou	University of the Aegean, Greece
Olivier Pereira	Université Catholique de Louvain, Belgium
Günther Pernul	Regensburg University, Germany
Andreas Pfitzmann	Dresden Technical University, Germany
Bart Preneel	Catholic University Leuven, Belgium
Kai Rannenberg	Frankfurt University, Germany
Peter Ryan	Newcastle University, UK
Ahmad-Reza Sadeghi	Ruhr University Bochum, Germany
Joseph Savirimuthu	Liverpool University, UK
Berry Schoenmakers	Eindhoven Technical University, Netherlands

Additional Reviewers

Roberto Araujo, Stefan Berthold, Sebastian Clauß, André Deuker, Stefan Duerbeck, Ludwig Fuchs, Sebastian Gajek, Yacine Gasmi, Jörg Gilberg, Jörg Helbach, Hugo Jonker, Andreas Juschka, Jan Kolter, Michael Kreutzer, Katja Liesebach, Olivier de Marneffe, Denis Royer, Hans Loehr, Tobias Scherner, Patrick Stewin, Martin Unger, Stefan Weber, Jan Zibuschka, Felix Zimmermann

Table of Contents

Improvements/Extensions of Existing Approaches

Code Voting

The Development of Remote E-Voting Around the World: A Review of Roads and Directions

Robert Krimmer[1], Stefan Triessnig[1], and Melanie Volkamer[2]

[1] Competence Center for Electronic Voting and Participation (E-Voting.CC)
[2] Institute of IT-Security and Security Law (University of Passau)
{r.krimmer,s.triessnig}@e-voting.cc, melanie.volkamer@uni-passau.de

Abstract. Democracy and elections have more than 2.500 years of tradition. Technology has always influenced and shaped the ways elections are held. Since the emergence of the Internet there has been the idea of conducting remote electronic elections. In this paper we reviewed 104 elections with a remote e-voting possibility based on research articles, working papers and also on press releases. We analyzed the cases with respect to the level where they take place, technology, using multiple channels, the size of the election and the provider of the system. Our findings show that while remote e-voting has arrived on the regional level and in organizations for binding elections, on the national level it is a very rare phenomenon. Further paper based elections are here to stay; most binding elections used remote e-voting in addition to the paper channel. Interestingly, providers of e-voting systems are usually only operating in their own territory, as out-of-country operations are very rare. In the long run, for remote e-voting to become a reality of the masses, a lot has to be done. The high number of excluded cases shows that not only documentation is scarce but also the knowledge of the effects of e-voting is rare as most cases are not following simple experimental designs used elsewhere.

1 Introduction

"While democracy must be more than [...] elections, it is also true [...] that it cannot be less," [2] former Secretary General Kofi Annan once said. Elections are the core element of democracy as a society's way to make decisions. Elections are the way to express how societies use technology and as new technologies have emerged and evolved, elections have changed accordingly. While there have been democratic structures in societies like India, the birthplace of democracy is attributed to old Athens in 507 BC [10]. From then on similar structures of direct democracy, bound by face-to-face societies, also developed in several places around the world like in ancient Rome [22], with the Vikings [32] or in the Cantons of Switzerland [34,19]. The next level of democracy developed with the creation of nation-states in the late 18th century with the need for representatives. This form of indirect democracy spread in three waves [24] from the United States and France around the globe to today's predominant role of democracy as a rule of government.

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 1–15, 2007.

The political scientist Robert Dahl classifies these developments as the first and the second transformation of democracy [9]. With it, democracy moved away from the old ideal of identity of the ruler and the ruled. Thus, the worldwide decrease in voter turnout and the rapid development of information and communication technologies like the Internet have led him and others to think about a third transformation - the development of the electronic democracy.

Positive visionaries like Grossman [17], Fuller [15] and Fromm [13] conceived the electronic republic with a new, more direct and pervasive form of democracy. Fuller anticipated even "electrified voting, [...] a mechanical mean[s] for nation-wide voting, daily and secretly, by each adult citizen." The more pessimistic view is taken by Golding [16] and Haywood [18], who foresaw a negative effect of new technologies for democracy, due to inequalities in information access. The experience with the transformational effect of the Internet on private (e-commerce) and public (e-government) sectors has strengthened the position of neutral researchers that foresee a similar transformational change for democracy (Bimber [3] and Leggewie & Bieber [30]), which will in the end develop a direct representation where representatives can be held more accountable by the electorate.

Either way, the development of an electronic democracy with transnational character [21] needs the further development of e-enabled instruments of democracy [20], i.e., e-initiatives, e-referenda and of course also e-voting instruments. Amongst them remote e-voting has received the largest attention, and it reached the national level in Estonia first. On March 3^{rd}, 2007 the Estonian national election offered the world's first legally binding remote electronic voting (e-voting) possibility [7]. With that event remote e-voting has finally reached the stage of international attention even though experts warned three years earlier in the SERVE report that the Internet is not ready for elections yet [25]. Most other nations are still in the phase of experimentation. To date most trials do not follow classical experimental setups [1] and are embedded in their national context [41] which makes it hard for comparison and learning from others.

This paper is the first attempt to conduct a state-of-the-art analysis [12] of 104 remote e-voting uses in the past twelve years to build knowledge on the future of voting. We analyzed the documentation in research articles, working papers and press releases of 104 e-elections conducted around the world. While we aimed for a representative sample, it is clear that the current cases cannot serve this purpose. Rather it gives an indication how remote e-voting has developed so far. In the following we will first give a theoretical background on remote e-voting, and then present the results of our review. Finally, we will discuss the findings and give our conclusions.

2 Theoretical Background

In this chapter we will explain what we mean by remote electronic voting and which methodology we used.

2.1 The Terminus Technicus Remote Electronic Voting and Its Variants

fi When talking about e-voting it is necessary to define the subject. The Council of Europe recommendations define electronic voting as "the use of electronic means in at least the casting of the vote" [35]. We first have to look at elections in a broad sense (for our purposes this includes e-referendums) and then concentrate on the implications of ICT usage therein.

The United Nations facilitated the agreement on the International Covenant on Civil and Political Rights [42]. Article 25 defines eight principles for elections that depict the whole electoral process: (i) periodic elections, (ii) genuine elections, (iii) stand for election, (iv) universal suffrage, (v) voting in elections on the basis of the right to vote, (vi) equal suffrage, (vii) secret vote, and (viii) free expression of the will of the voters. Suksi [40] groups these principles into a cycle consisting of three periods:

1. Pre-Electoral Period: This is the time from calling an election until the actual start of the polling.
2. Electoral Period: This is the actual Election Day where the vote casting takes place.
3. Post-Electoral Period: This is the time during which the results are announced and a new election is called.

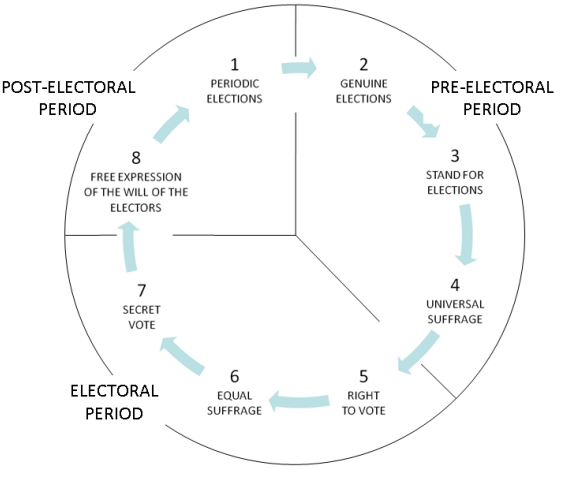

Fig. 1. The electoral cycle [40]

The electoral process usually takes place at the polling station and is supervised. This can be referred to as voting at presence. But there is also the possibility of remote voting. The criterion to differentiate those two is if an election commission supervises the act of voting or not [27]. At current elections

the voter comes to the polling station where the election commission checks the identity and eligibility and ensures the voter's anonymity when casting the ballot. When the election has finished the election commission counts the votes. With remote elections the identity and the right to vote is checked beforehand or remotely and the voter has to make sure that his anonymity is not compromised. This raises questions of voter coercion and vote buying [29].

According to the dimensions of medium and place of voting the systems can be assigned to six basic groups. The medium hand is characterized by its inherent need of presence and is limited to a certain number of people and does not allow for voting in an uncontrolled environment. In modern institutionalized elections this medium is very seldom used. Most modern day elections use paper as a medium of choice. Polling station voting using paper ballots is characterized by the controlled environment and the usage of paper as a medium. Postal voting also uses the medium paper, but provides no controlled environment. If the ballot is cast electronically one can differentiate between voting machines that are placed in the controlled environment of a voting station and remote electronic voting that also uses an electronic channel as a medium, but provides no controlled environment. Table 1 gives an overview for detailed information see [43].

Table 1. Forms of electronic voting

Environment / **Medium**	Controlled	Uncontrolled
Hand	In-Person	-
Paper	Polling Place	Postal Voting
Electronic	Voting Machine	Remote Electronic Voting

It is possible that one election uses more than one form of voting. Critical from the operational viewpoint is if more than one channel is allowed and if paper and electronic channels have to be combined. When counting the votes the system must ensure that multiple voting in different channels is not possible. One has to make sure that the individual results of the channels are combined in such a way that the end result is correct. For the time being, democracy theory and constitutional law (requirement of universality) require additional paper channels as long as not everyone has the skill and access to the Internet, thus remote e-voting can only be an optional channel in legally binding elections for the time being.

Remote e-voting can take place at elections of diverse levels of attention. We differentiate five different levels determined by political importance, legal commitment, and parallel testing. The political importance is defined by Lijphart [37] as such that the first and the second level elections are politically binding which means they are regulated by law and the results of the elections have consequences. The most rigid legal framework is found with first level elections like presidential or parliamentarian. On the second level less important political elections can be found. Typical elections for that level would be local elections.

Elections of lesser importance, because of their lesser political impact like (student) union elections or elections in corporations, can be considered as the third level. These tend to have fewer rules on how the election has to be conducted. Still some kind of outcome is dependent on the result of the election. Critical for all of them is that they have to fulfill certain rules so the outcome of the election can be binding and some kind of action can be derived.

This leads to another classification of elections. A test is an election that's sole purpose is to test the system. Such tests are often conducted in an early stage of the development of a system and their sole purpose is to test the system. A logical next step is to simulate an election and test the system parallel to a binding one. The aim of such a test is to trial the system under realistic conditions and the results of which are not legally binding. These five categories build the five levels of elections:

Table 2. Levels of elections

Levels	Leg. Binding	Org. Binding	Non-Binding
1st Level: national	X		
2nd Level: regional, local	X		
3rd Level: org., assoc., companies	(X)	X	
4th Level: shadow, parallel			X
5th Level: technical test			X

[illegible] The basic problem of electronic voting is how to solve of the unequivocal identification of a voter and at same time being able to guarantee anonymity with a secret ballot casting [31].

[illegible] For identifying a voter three basic criteria can be used to differentiate the technologies: (i) knowledge, (ii) possession, and (iii) properties. A fourth possibility is a combination thereof. These identification technologies are used in remote e-voting:

1. *[illegible]* The identification relies on the voter knowing a secret.
2. *[illegible]* The voter possesses something that identifies him/herself.
3. *[illegible]* The voter him/herself with his/her individual biometric properties identifies him/herself. A reader for the biometric feature is needed.
4. *[illegible]* The voter knows a secret that in combination with the possession of the card identifies him. Or a property pattern of the voter is stored on the smart card that is checked against the voter's property when casting a ballot -either way, a reader for the smart card is needed.

[illegible] Critical for a voting system is the question of guaranteeing anonymity. There have been many articles written to categorize and cluster protocols guaranteeing anonymity [26,37,33,23]. While the criteria used in these papers are

very sophisticated, in practice a simpler and more distinctive criterion is time [39]: At which point in the electoral cycle is secrecy (anonymity) established?

1. In the Pre-electoral Period: Anonymity is established in the pre-election period by the organizing institution. The most common implementation of such a system uses transaction numbers (TAN). These numbers are generated centrally and a scratch-field is applied. Then in a second step the voter's address is applied and sent to the voter who can use the number anonymously for exactly one vote.
2. During the electoral period: With this method the anonymity is established during the vote casting procedure. It can either be done by separating the servers in an identification and ballot box server or by blind signatures; the most common implementation of Chaum's blind signature [36] is in the Fujioka et al. algorithm [5]. The process can be explained as follows: the voter fills out his/her ballot sheet, then puts it in an carbon-copy envelope. The voter then signs another envelope with his/her personal signature and inserts the carbon-copy envelope and sends the package to his/her register. They check the voting eligibility based on the voter's signature, then sign the carbon copy envelope and return it to the voter. The voter opens the cc-envelope and has a signed ballot sheet (due to the carbon copy) without the voter's register ever having seen the ballot sheet. Finally she returns the ballot sheet to the ballot box and has thereby cast a valid vote anonymously.
3. In the post-electoral period: In this case the anonymity is established after the end of the election day, when the votes can still be identified but the count can only be conducted together meaning the content of a single vote is never released. The most common implementations use homomorphic encryption like the Schoenmakers algorithm [14] or hardware security modules like the Estonian system [38]. Provider. To conduct an electronic election is a complex undertaking and is usually operated by a consortium. We identified the provider that was critical or characteristic for the whole system. Of special interest was in which country the provider operated and how much experience the company had.

One important criterion for assessing e-voting use is how many votes are cast. Looking at the sample we found it useful to group the elections into three size groups. The first group (A) contains all elections with more than 30,000 votes. The middle group (B) contains elections with a number of e-votes between 3,000 and 30,000. The last group (C) consists of small elections with a number of e-votes smaller than 3,000.

2.2 Research Methodology

Conducting a review can be organised in many ways; the approach we selected follows the handbook of review synthesis [6] which proposes five phases: (i) problem description, (ii) literature research, (iii) literature analysis, (iv) analysis, and (v) presentation.

Table 3. Criteria to categorize remote e-voting

Criterion	Category
Level	National - Regional - Association - Shadow - Test
Channels	Electronic - Paper and Electronic
Identification	Username/PWD - TAN - Signature - Biometric
Anonymity	Pre-electoral period - Electoral period - Post-electoral period
# Votes	# > 30,000 - 30,000 > # > 3,000 - # < 3,000

(i) The goal of this review was to conduct a review on the progress of remote electronic voting. (ii) To use as sources we consulted research articles, system documentation, whitepapers, technical reports, and even press releases if necessary. As remote electronic voting is a very new topic for the general public, often more than one source had to be consulted to gain a complete picture. Not surprisingly research articles usually gave a better insight on the project setup and system description while lacking actual election related data. This was where we consulted press releases. To find the appropriated sources we used a network of experts around the world that were invited to provide data or point to relevant documents. We provided them an online questionnaire on a public website to identify relevant elections. Because of the multitude of sources the data had to be consolidated. That makes it difficult to find common ground, so we needed to add an extensive array of integration work. (iii) The criteria that were developed in the previous chapter were used to characterize the elections. (iv) The collected data was then entered it into a database for analysis, and (v) then presented and discussed in the following chapters.

3 Results

In total we identified 139 elections in 16 countries within the time period of 1996 to the 30^{th} of April 2007 where remote e-voting occurred. For the analysis we needed a minimum amount of information about every election. We had to eliminate 35 (!) elections in total. Three elections were excluded from analysis because of missing data about voters and turnout. The largest exclusion reason was for not having system documentation available, which applied to thirty elections. Without the documents we could not assess which forms of identification or anonymity were used. Finally, two could not be included at all because we lacked information on the voter data and on the used system. In total we had 104 fully documented elections which we could include in the following analysis. These elections were held in 13 different countries on three continents; two elections were held transnationally. The first election was held in 1996 in Finland and the last in 2007 in Estonia. The following table shows the distribution of all elections over time and by country. From the analysis, excluded elections are put in brackets.

The countries with the most elections were Germany (30), Switzerland (24) and the United Kingdom (19). Surprisingly the United States has just 2 publicly documented elections.

Table 4. Number of elections per year and country included (excluded) in review

Year	AG	AT	AU	CA	CH	DE	EE	ES	FI	FR	NL	PT	SE	UK	US	WW	Total
1996									1								1
1998						1											1
1999						1(1)			1						1		2(1)
2000		1				2(3)									1(1)	1	5(4)
2001					(3)	4(1)							1				5(4)
2002					(2)	2 (1)		(1)		3				5			10(4)
2003		1			2	3		1(2)		2				14			23(2)
2004		2			7	4(2)		2(3)			2					1	18(5)
2005	1				10	3(3)	2	2(3)				1					18(7)
2006		1	(1)	1	4	9		(4)		1	1						17(5)
2007			(1)		1	1	1			1							4(1)
????								(2)									(2)
incl.		5		1	24	30	3	5	2	7	3	1	1	19	1	2	104
(excl.)	(1)		(2)		(5)	(11)		(15)							(1)		(35)

We will walk you through the process of classifying elections with the example of the 2007 parliamentary elections in Estonia. The election was on the national level and was legally binding. This places the election into level 1 of the 5 levels. It was also a multi-channel election that offered both paper and remote e-voting channels. Voters could cast their vote electronically over the internet before Election Day or at local polling stations on or before Election Day on paper. The voters could use the remote e-voting system with their national ID card, a smart card which bears a digital signature. The vote is first encrypted using the public key of the ballot box, and then signed by the voter with her private key. To count the votes Estonia uses a hardware security module for hidden result calculation which means anonymity is established in the post-electoral period. The provider of the system was Cybernetica AS, which is of Estonian origin. Approximately 940,000 people were eligible registered voters and 30,275 cast their votes electronically. This places the election in group A of large elections.

The other elections were categorized in the same way. The result of the systematization is depicted in table 5 and is described below.

With 38 cases the group of 2nd level elections is the biggest. The 3rd level is the second-largest with 30 elections. Of all binding elections the group of national elections is with four instances the smallest (once each in Estonia and Switzerland, twice in the Netherlands). The group with shadow elections has 27 elections and only five elections had a sole test purpose. Interestingly the legally binding elections attribute for over 40% of the cases.

In one third of the cases the remote voting channel was the only method to cast votes. For the majority (65 cases) of the elections, e-voting was just an additional channel to the traditional paper way.

Identification: With 84 elections the most favourite way of identifying voters, by far, was the TAN-system. 15 elections used signature cards and just 4 elections

Table 5. Number of elections per year and country included (excluded) in review

Criterion	Category				
Level	National (4; 3.8%)	- **Regional** **(38; 36.5%)**	- Association (30; 28.9%)	- Shadow (27; 26%)	- Test (5; 4.8%)
Channels	Electronic (39; 37.5%)	- **Paper and Electronic** **(65; 62.5%)**			
Identification	Username/PWD (4; 3.9%)	- **TAN** **(84; 81.5%)**	- Signature (15; 14.6%)	- Biometric (0; 0%)	
Anonymity	**Pre-electoral period** **(53; 50.9%)**	- Electoral period (29; 28.2%)	- Post-electoral period (21; 20.4%)		
# Votes	# > 30,000 (9; 8.7%)	- 30,000 > # > 3,000 (30; 28.9%)	- # < **3,000** **(65; 62.4%)**		

used a relatively insecure username and password system. Biometric systems were not used at all.

Anonymity: In two-thirds of the investigated remote e-voting elections the anonymity was established before Election Day using organizational pre-registration. The second most common way is to establish it during the electoral period, which was used in 28.2% of the cases. The use of establishing anonymity after the election was used in 20.4% of the cases.

One election did not fit the categorization in the field of identification and anonymity because the identification was done based on IP-address and anonymity could therefore just be guaranteed organizationally.

The elections with remote e-voting have a large span width between the largest (130,000) and smallest (54) number of voters. Most elections were rather small, as 65 elections had fewer than 3,000 votes cast. 28.9% of the elections had between 3,000 and 30,000 voters. In the largest group with over 30,000 votes only 9 elections could be found.

In total 25 different providers organized the analysed elections. Four of them account for 54.8% of all conducted elections, while the other 45.2% were distributed amongst 21 providers. Most providers (76%) have experience only in their home country; the six who have operated elections outside their home country have done so in a maximum of three foreign destinations. Only one provider has operated solely abroad which is due to the fact that it is homed in the US but also has a strong base in European countries.

4 Discussion

Starting with the reported findings in the previous chapter we will discuss the results here more closely. The "idea" of collecting all elections was very ambitious. 1st level and most 2^{nd} level documentation is publicly available. Most of the time it is not in one place but with enough work the information can be

gathered. For elections on the third level most of the time public information is hard to get. We know that there are a lot of elections in the US in the private sector but simply could not get public documentation for them.

Everybody wants to sell a success story. This is especially noticeable when looking at turnout data. The most inconvenient low numbers simply get left out. The problem of selective information is not just a problem with result numbers but with information about elections in general. A language and regional bias is noticeable and also inherent in the method of experts referring to experts and resources. Nearly all papers and documentation just deal with single cases. There are very few comparative sources. Some initiatives can be found, but nothing comprehensive.

Generally it is hard to maintain data quality. The problems result from combining multiple sources that use different wording, are incomplete, and even contradicting.

A broader constant process would be needed. The US and Asia can surely contribute to the process. Experts are asked to leave their box and overcome their bias. A start would be the 30 elections that had to be excluded because of missing technical system documentation.

The number of elections using remote e-voting has risen during the time span in our review. Interestingly most of the cases took place in the new millennium with a heap in 2003, and have maintained at that level since then. Further, the number of countries acting on e-voting is rising as well. Still it has to be mentioned that the average cycle for political elections is 4-5 years, which also limits the number of possible legally binding e-voting uses. We also noticed a strong bias of remote e-voting in Europe, where 100 of the 104 cases are located. This is of course due to the fact that Europe with its large number of countries also inherently has the largest number of elections to conduct. Furthermore the biggest potential of remote e-voting - to conduct trans-national elections - has not yet really been taken advantage of. Only two elections in that area have been noticed so far. This probably also deals with the fact that these elections could only happen on a 3rd level as the potential candidate for this - the European Union - has no mandate for elections yet and cannot make legislation for this as of now.

We were surprised that 40% of the conducted elections were legally binding (1^{st} and 2^{nd} level). A large stake can be attributed to the pilot series at the local level in 2002 & 2003 in the United Kingdom. On the national level the number is much smaller and has happened only in three countries (Estonia, Netherlands and Switzerland). In most countries to use remote e-voting channels, laws or even the constitution have to be changed, which makes remote e-voting very unlikely to happen on the go. It needs a strategic intention of the government for this. On the third level with (not legally) binding elections we expected more cases, but instead they make up only 29% of the total number. This could relate to a lack of interest in publishing the experiences with remote e-voting. Reasons could be only a small interest of the public, or that it has

been conducted more than once already. In the field of non-binding elections, i.e., the area of testing a system, it is clear that most cases took place in parallel to a real election and only few are pure functionality tests or fictional elections. The reason for this is the problem of motivating the voters - why should they participate?

A lot of attention should be placed on the results in the field of identification. The numbers showed very clearly that the ID of choice for electronic voting is a TAN. It is easy to handle as voters know it from lottery tickets. In addition it is also cost effective as no reader is needed. Further the TAN is a good way for the election organizers to make project marketing. The most secure way - signature cards - obviously has a problem with usability and is too costly.

Similar to the case with the identification we found that most election organizers (71.3%) choose algorithms that keep the establishment of anonymity in their premises, i.e., either before or after Election Day. This has to do with the fact that in these algorithms the least number of calculations is necessary on the side of the voter which means in consequence that the voting procedure requires less additional software like java programs or applets and can run in an ordinary browser. Establishment during the electoral period was used in 28.2% of the cases.

If we check the use of multiple channels in combination with the five levels a clear pattern emerges. 99% of all legally binding elections at the national and regional levels have at least one paper channel parallel to the electronic channel. In the 3^{rd} level 58% use only electronic channels and 42% also use paper and electronic channels at the same time. The 4^{th} level excludes per definition paper-based channels and the 5^{th} level just uses electronic channels.

When looking at numbers for votes cast one can clearly see that electronic elections are still an emerging field. Systems are gradually tested starting with smaller numbers. But in absolute figures all of these elections are not comparable to traditional elections. The biggest legally binding election to date - the Arizona State Democratic Preference Primary in March 2000 - had around 40 thousand votes cast.

Only four providers organized the majority of elections. These are also the providers which organized elections in different countries. The rest is distributed among 21 providers which in most cases just operate in their home country. This is most probably explained by the lack of trust in foreign companies and fear of them controlling such a core element of democracy.

5 Conclusion

With a field being around for 12 years, a review of the collected experience was long needed. A review of the conducted e-elections on a structured basis was a

challenge due to the fragmented characteristic of the available information. Our sample of 104 cases covers 12 year, 3 continents and 14 countries. In general data quality is the biggest obstacle to overcome.

Our research shows that although there have been four legally binding top-level remote e-voting elections the field is not mature yet. The best indicator is the relatively small size of the cases. 62% of the elections have less than 3,000 voters and only 8.7% have more than 30,000. These numbers are far from any traditional election.

The obvious target area foreseen by the visionaries - citizens living abroad and transnational elections - was the focus of only seven elections.

Conducting e-elections needs a technical provider who usually is an IT-company. Interestingly they operate only in their home country. There seems to be resistance in engaging companies from abroad.

For the implementation selecting the right identification and anonymity schema is crucial for success. Here most cases selected a combination of TAN and pre-electoral establishment of anonymity. The information of a theoretically more secure signature and establishment during vote casting falls back in adoption most probably because of additionally needed infrastructure. However the Estonian example shows that legally binding remote e-voting with signature smart cards is possible.

Handling multi channels involving paper and electronic vote casting does not seem to be a problem. On the contrary, 99% of all legally binding elections offered remote e-voting only in addition to paper-based vote casting.

In the future research has to focus on its role in understanding and learning from what has been done so far. In this way, any academic involved in remote e-voting should follow basic academic styles. Not only does that mean the experiments should follow basic experimental designs but also documentation should be comprehensive, analytic and comparable. Based on existing approaches [8,4] academics should develop a guideline on how to properly document e-voting uses, similar to election observation reports [28,11].

To make this research more valuable it should be accessible by third parties in a public database. This would help readers learn from the results and also gain further insights in projects not included in this review.

It would also be interesting to deepen the analysis of this material available especially in the field of technology following a longitudinal approach. Here the development could deliver interesting insight into the adoption of identification and anonymity technologies.

Overall remote electronic voting has not reached the maturity to be applied in large-scale elections of major importance. More research needs to be put into the effects, outcomes and security of remote e-voting. Documenting the experience, as has been done here, is a first step to build up a research strategy.

The authors would like to thank Nadja Braun, Thomas Buchsbaum, Letizia Caporusso, Alexander Schellong for comments on an earlier version of the text.

References

1. Alvarez, R.M., Hall, T. (eds.): Point, Click, & Vote. Brookings Institution Press, Washington, D.C (2004)
2. Annan, K.: UN Secretary General Kofi Annan's Closing Remarks to the Ministerial on June 27th 2000 (2000), accessed on 2007-05-29, http://www.demcoalition.org/pdf/un_secertary_gen_kofi_annan.pdf
3. Bimber, B.: Information and political engagement in America: The search for effects of information technology at the individual level. Political Research Quarterly 54(1) (2001)
4. Buchsbaum, T.: Questions and challenges of e-voting, and attempts to find answers and solutions (accessed on 2007-08-19) (2005), http://www.bmeia.gv.at/upmedia/1552_buchsbaum_e_questions_challenges_2.pdf
5. Chaum, D.: Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. Communications of the ACM 24(2), 84–88 (1981)
6. Estonian National Electoral Committee. General Description of the E-Voting System (2004), http://www.vvk.ee/elektr/docs/Yldkirjeldus-eng.pdf accessed on 2005-10-20
7. Estonian National Electoral Committee. Parlamentary Elections (2007), http://www.vvk.ee/r07/paeveng.stm
8. Cooper, H., Hedges, L.V.: The Handbook of Research Synthesis, Russel Sage Found, New York (1994)
9. Dahl, R.A.: Democracy and its critics. Yale University Press, New Haven + London (1989)
10. Dahl, R.A. (ed.): On Democracy. Yale University Press, New Haven, London (1998)
11. Eriksson, A.: Handbook for European Union Election Observation Missions, Sida, Stockholm (2002)
12. Fettke, P.: State-of-the-Art des State-of-the-Art. Eine Untersuchung der Forschungsmethode "Review" innerhalb der Wirtschaftsinformatik. Wirtschaftsinformatik 48(4), 257–266 (2006)
13. Fromm, E.: The Sane Society, Rinehart, New York (1955)
14. Fujioka, A., Okamoto, T., Ohta, K.: A Practical Secret Voting Scheme for Large Scale Elections. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 244–251. Springer, Heidelberg (1993)
15. Fuller, R.B.: No More Secondhand God (late Night, April 9, 1940). In: No More Secondhand God - and other writings, 4th edn., pp. 3–36. Southern Illinois University Press, Carbondale (1963)
16. Golding, P.: World Wide Wedge: Division and Contradiction in the Global Information Infrastructure. Monthly Review 48(3), 70–85 (1996)
17. Grossman, L.K.: The Electronic Republic: Reshaping Democracy in the Information Age, Viking Penguin, New York (1995)
18. Haywood, T.: Info-Rich / Info-Poor: Access and Exchange in the Global Information Society, Bowker-Saur, East Grindstead (1995)
19. Head, R.C. (ed.): Early Modern Democracy in the Grisons. Cambridge University Press, Cambridge (1995)
20. Heindl, P.: Elektronische Demokratie- "Dienstleistungen" des Staates: E-Voting, E-Legislation und E-Participation. In: Prosser, A., Krimmer, R. (eds.) E-Democracy: Technology, Law and Politics, vol. 174, pp. 175–188. OCG Verlag, Vienna (2003)
21. Held, D.: The Transformation of Political Community: Rethinking Democracy in the Context of Globalization. In: Shapiro, I., Hacker-Cordn, C. (eds.) Democracy's Edges, pp. 84–111. Cambridge University Press, Cambridge + New York (1999)

22. Held, D.: Models of Democracy, 3rd edn. Polity Press, Cambridge, Malden (2006)
23. Horster, P., Michels, M.: Der Vertrauensaspekt in elektronischen Wahlen. In: Horster, P. (ed.) Trust Center, Vieweg, Braunschweig, pp. 180–189 (1995)
24. Huntington, S.P.: The Third Wave. Democratization in the Late Twentieth Century. In: The Third Wave. Democratization in the Late Twentieth Century, University of Oklahoma Press, Norman (1993)
25. Jefferson, D., Rubin, A., Simons, B., Wagner, D.: A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE) (2004)
26. Kofler, R., Krimmer, R., Prosser, A.: Electronic voting: algorithmic and implementation issues. In: HICSS36 (2003)
27. Krimmer, R.: e-Voting.at - Elektronische Demokratie am Beispiel der österreichischen Hochschülerschaftswahlen. Diploma thesis, Vienna University of Economics and Business Administration (2002)
28. Krimmer, R., Triessnig, S.: Database of E-Voting Uses Worldwide (2007), accessed on 2007-01-31, http://DB.e-voting.cc/
29. Krimmer, R., Volkamer, M.: Bits or Paper? Comparing Remote Electronic Voting to Postal Voting. In: Andersen, K.V., Grönlund, A., Traunmüller, R., Wimmer, M. (eds.) Workshop and Poster Proceedings of the Fourth International EGOV Conference 2005, pp. 225–232. Universitätsverlag Rudolf Trauner, Copenhagen (2005)
30. Leggewie, C., Bieber, C.: Interaktive Demokratie. Aus Politik und Zeitgeschichte 41(42), 37–45 (2001)
31. Lijphart, A.: The Problem of Low and Unequal Voter Turnout - and What We Can Do About It. Technical Report 54, Institut für Höhere Studien (IHS), February 1998 (1998)
32. Lars, L.: The Vikings in History and Legend. In: Sawyer, P. (ed.) The Oxford Illustrated History of the Vikings, pp. 225–249. Oxford University Press, Oxford, New York (1997)
33. Mitrou, L., Gritzalis, D.A., Katsikas, S., Quirchmayr, G.: Electronic Voting: Constitutional and Legal Requirements, and Their Technical Implications. In: Gritzalis, D.A. (ed.) Secure Electronic Voting, pp. 43–60. Kluwer Academic Publishers, Boston + Dordrecht (2003)
34. Silvano, M.: Die schweizerische Landsgemeinde-Demokratien, Haupt, Bern (1987)
35. Council of Europe. Legal, operational and technical standards for e-voting. Recommundation Rec(2004)11 and explanatory memorandum. Council of Europe, Strassbourg (2004)
36. Puiggali, J.: Introduction to Remote Electronic Voting (2007), http://www.scytl.com/docs/pub/science/Master-ESCERT-07-voting-En.pdf accessed on 2007-05-29
37. Schlifni, M.: Electronic Voting Systems and Electronic Democracy: Participatory E-politics for a New Wave of Democracy. Dissertation, Technische Universität Wien (2000)
38. Schoenmakers, B.: A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 148–164. Springer, Heidelberg (1999)
39. Warren, D., Smith, W.D.: Cryptography meets Voting (2005), http://www.math.temple.edu/∼wds/homepage/cryptovot.pdf accessed on 2005-08-07
40. Suksi, M.: The Electoral Cycle: On the Right to Participate in the Electoral Process. In: Hinz, V.U., Suksi, M. (eds.) Election Elements: On the International Standards of Electoral Participation, pp. 1–42. Institute for Human Rights, Abo Akademi University, Turku/Abo (2003)

41. Svensson, J., Leenes, R.: E-Voting in Europe: Divergent democratic practice. Information Polity, 1–2, 3–15 (2003)
42. UNHCR. International Covenant on Civil and Political Rights (1966), accessed on 2007-06-11, http://www.unhchr.ch/html/menu3/b/a_ccpr.htm
43. Volkamer, M., Krimmer, R.: Die Online-Wahl auf dem Weg zum Durchbruch. Informatik Spektrum 29(2), 98–113 (2006)

Remote Voting Schemes: A Comparative Analysis

Jordi Puiggali and Victor Morales-Rocha

Scytl Secure Electronic Voting
Tuset 20 1-7, 08006 Barcelona, Spain
{jordi.puiggali,victor.morales}@scytl.com

Abstract. Some governments initially introduced postal voting as a way to facilitate overseas and absentee voter's access to the electoral process. However, reliability issues that are part of postal voting have helped to introduce new remote voting channels based on electronic means. In the United States, electronic voting channels based on fax, email or Internet, are or have been used in binding elections. In this paper we identify which are the current remote voting channels used in the United Sates for overseas voters binding elections. We also identify which are the main criteria requirements used to evaluate the implementation of these remote voting channels, and provide a general comparison of the fulfilment of these requirements by these remote voting channels.

Keywords: Remote voting, remote electronic voting, postal voting, Internet voting.

1 Introduction

In many countries, governments are using postal voting to allow voters to cast their votes remotely. However, reliability issues regarding the postal transportation of election material has motivated governments to study, and in some cases, implement new alternative models of remote voting based on electronic means [1]. Countries such as Switzerland [2], United Kingdom [3] and Estonia [4] are implementing remote electronic voting in their binding elections. Other countries, such as Australia [5], are using postal voting and also evaluating the use of remote electronic voting in the near future. In the United States, some States support remote electronic voting channels based on fax and/or email as alternative methods to postal voting for their overseas voters [6].

In general, countries are implementing remote electronic voting since it is perceived as a more reliable transmission channel than postal voting. Voters have less restrictions for casting their votes on time (e.g., voters do not need to deal with delays on the delivery of their votes), and also know if their votes reach the election officer practically at the same time they cast their votes. Furthermore, remote electronic voting also improves the accessibility of the voting process and, in some implementations such as Internet voting, helps voters to cast a vote without making involuntary errors that could invalidate it. However, concerns

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 16–28, 2007.

related to the security of these platforms are delaying or even preventing their adoption in some places.

In 2005, Krimmer and Volkamer [7] introduced a framework used to compare the postal voting and Internet voting platforms used in GI's chairman election (a German Association devoted to the promotion of the Computer Science). However, this framework is focused to evaluate specific projects and does not include a specific analysis of both platforms. In this document we identify the remote voting channels used now a day in the United States for overseas voters. Then, we implement a general comparison of them. Even though Internet voting was cancelled in 2004 due to a technical report [8], we included this channel in our analysis in order to evaluate its benefits and drawbacks. The aim of this comparison is to provide a framework for the evaluation of the security, usability and election management of remote voting methods. It is also an objective of this comparison to evaluate the current situation of the voting methods used in the United States for overseas voters.

The organization of this document is as follows. Section 2 shows a classification of remote voting schemes and describes each of these channels. In section 3 an evaluation criteria for remote voting schemes is described. In section 4 we do a comparative analysis of the distinct voting channels. Finally, section 5 concludes.

2 Remote Voting Schemes

In the United States, there are different ways to cast a ballot remotely, being postal voting the most used remote voting channel now a days. In the case of overseas voters, a second group of remote voting schemes based on electronic means is also supported. This group is composed of email and fax voting schemes. Internet voting is another remote electronic voting scheme that was used in the United States for overseas voters. It was used in 2000 [9] when The Federal Voting Assistance Program (FVAP) conducted a pilot project for the presidential election. However, as we have previously mentioned, the Internet voting project was cancelled in 2004 due to the results of the report "A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)" [8]. Nevertheless, we are considering Internet voting in this study since it is currently under evaluation for future implementations. A single classification of the voting channels that we will analyze is shown in figure 1.

Remote voting has some advantages in regard to conventional in person voting. For instance: greater convenience for voters, particularly for those with limited mobility or those living in remote areas. Furthermore, remote voting allows a higher non-resident participation.

On the other hand, the main concerns for all remote voting schemes are those related to voter's privacy and coercion, regardless if they are based on paper or electronic ballots [7]. Being a remote and most of the time an uncontrolled environment, voter is exposed to family voting, manipulation, coercion, and lack of privacy at time to cast a vote.

The following sections describe the different channels considered in this analysis, including the main particular concerns about each channel.

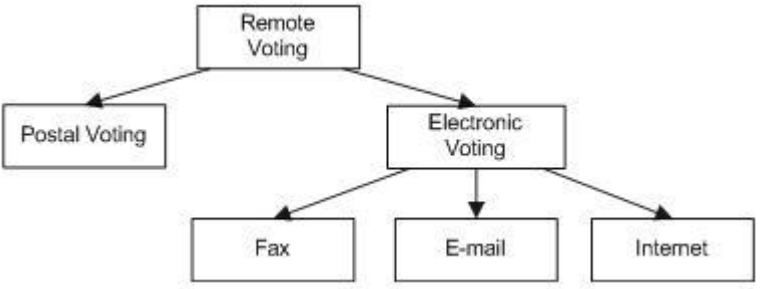

Fig. 1. Classification of remote voting channels

2.1 Postal Voting

According to Matt Qvortrup [10], "postal voting is the use of the postal ballot as a more or less comprehensive alternative to attendance voting. Instead of having a day on which voters attend polling booths to cast their votes, they receive a ballot paper by post and then have a period in which to return their vote by mail before Election Day".

Voters usually introduce their votes into an unmarked envelope. This envelope is then introduced into a second envelope with a voter certificate (which identifies the voter). The second envelope is sent to the election official through postal mail. Usually voters also sign a form that is included in the outer envelope with his/her personal data that could be used as proof of identity.

As described in [11] and [1], in postal voting there is a risk that election material or cast votes are not received on time to be counted by authorities. This risk is due to possible problems in the postal service. These problems are usually related to the delay in delivering the ballots, either in the transportation of blank ballots from the election official to voter or in the transportation of returned voted ballots. Therefore, due to such delays, unintentional disfranchisement could be present during an election. That is the main problem in respect of this remote voting channel. An example of the reliability problems of postal voting in contrast to alternative electronic channels can be found in [12]. In this experience over 23.000 Spanish overseas citizens were invited to participate in a non-binding remote electronic voting in parallel to the binding channel (postal voting) for the 2003 Elections to the Parliament of Catalonia. Due to delivery delays of the election material in one of the Countries (Mexico), most of its resident voters could not cast their postal vote on time, and therefore the number of non-binding Internet votes exceeded in more than 200 % the number of valid postal votes.

In addition to reliability issues, postal votes could also face integrity risks during their transportation. These votes can be manipulated or eliminated during their transport.

To mitigate such risks there are proposals based on the use of tracking codes [13] in the vote. However these proposals only allow detecting reliability problems but do not provide a solution to vote manipulation.

2.2 Fax Voting

Fax voting, is supported by 24 States of United States [6] and consists on transmitting the vote by fax to a pre-assigned number [14]. This channel is mainly supported as a contingency measure in case voters cannot cast their votes on time. In parallel, Election Officials also recommended sending the same vote by postal mail.

The main advantage of fax voting is that it solves the uncertain reception problems of postal voting systems. Voters know if their votes have been received by the Election Officials if the transmission of the fax finishes successfully.

The main issue of fax voting is related to vote secrecy. The vote is transmitted in clear through an unprotected channel along with the voter identity. Therefore the contents and the voter intent are known when received or could be intercepted during the transmission. Election Officials, in an attempt to increase the election privacy, put the vote into an envelope once the voter right to vote is verified. However this measure only protects the vote after being enveloped.

For this reason, States allowing fax voting require voters to sign a secrecy waiver where voters agree that they are waiving the right to secrecy of the ballot.

2.3 Email Voting

The Email voting channel is supported by 7 US States [6]. This channel requires voters to send an email with a scanned version of their votes (PDF formatted) attached. This email is sent to the Local Election Official email address. If the vote is accepted, the emailed vote (PDF) is printed by the Election Official and put into an envelope to keep it safe until it is counted. As in fax voting, voters are required to sign a secrecy waiver to agree that they are waiving the right to secrecy of the ballot [14]. Therefore it shares similar secrecy concerns as fax voting since the ballot contents are sent in clear along with the voter identity.

In addition to fax concerns, email voting poses a new risk: the use of public email relays for sending the votes. This could facilitate the eavesdropping of the votes or manipulation of contents. Furthermore it does not provide the same channel reliability as fax voting, since the reception of the email can be delayed by the intermediate email relays. However it is considered more reliable than postal voting, since voters can detect these problems earlier than postal voting. In such a case they can try to vote again or use another channel.

2.4 Internet Voting

In the Internet Voting Report of the California Internet Voting Task Force [15] the following definition is given: “an Internet Voting System is defined as an election system that uses electronic ballots that would allow voters to transmit their voted ballot to Election Officials over the Internet”. This voting channel was used by overseas voters of 4 States during the 2000 presidential elections [9]. However, the adoption of Internet voting for overseas voters was stopped in 2004

due to security concerns, after the publication of the "A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)" report [8]. In this report, the authors highlighted that the use of unsecured voter PCs and the intrinsically insecure design of Internet poses important privacy, integrity and reliability risks to the voting process. Privacy and integrity risks are mainly related to the introduction of virus or malware in the voter's PC. Reliability risks are related to denial of service, spoofing and man in the middle attacks that are commonly used throughout the Internet. Furthermore, the report also highlights that such attacks can be easily scaled, increasing the impact of the attacks.

The attacks described above are exclusive to Internet Voting systems (and any other voting platform that uses voters PCs and Internet). However, it is important to note that the other voting channels described here are also susceptible to specific attacks which could have a large scale impact on the election. For example, postal ballots can be intercepted during transportation and therefore they can be manipulated or even eliminated. Therefore one of the objectives of this study is to try to find similarities on the risks posed by each channel, to facilitate the evaluation of severity of these risks among the channels.

On the other hand, Internet voting shares the same advantages as the previous electronic voting systems. Furthermore, the fact that the voters use an interactive environment to cast their votes provides additional advantages such as the prevention of casting invalid votes due to involuntary voter errors. It also allows the implementation of cryptographic voting protocols that solve the secrecy and authenticity issues of previous remote voting schemes. Therefore, the use of such cryptographic measures is of paramount importance to reduce the risk of impact on Internet voting. Since there is not a standard model of Internet voting platform in the United States, we used the voting platform proposed by Riera [16] as a reference for this scheme. This proposal is being used to conduct Internet binding elections in several countries such as Switzerland (Neuchatel canton), United Kingdom and Philippines.

3 Evaluation Criteria

As previously introduced, each voting scheme has its own security risks and characteristics. Therefore framework must be defined to facilitate the comparison among these schemes. Voting system requirements are described in different works, e.g. [17] and [18]. This section provides a summary of such requirements, where they are classified as security, usability and election management. We will use these requirements as evaluation criteria for remote voting channels.

3.1 Security

From a security point of view, the following requirements must be fulfilled.

Eligibility. Only authorized voters should be able to vote. This means that the channel must provide a robust way to remotely identify voters and detect

any impersonation attempt. One of the main issues of remote voting is that voters cannot be identified in person. Therefore we can distinguish two different levels of impersonation: voluntary and involuntary. Involuntary impersonation is related to the impersonation of the voter without his/her knowledge (e.g., the theft of the voter credentials required to cast a vote). Voluntary impersonation requires the participation of the voter, who cooperates with the person that will impersonate such voter by providing his/her voting credentials. With the aim to simplify the comparison, we considered the risks of voluntary impersonation in the coercion and vote buying resistance security requirement. Therefore we only evaluated the risks of involuntary impersonation in this eligibity requirement.

Privacy. The voting system has to protect voter privacy, concealing the relation between voter and his/her cast vote, and ensuring that the voter's choice will remain anonymous. This requirement must be fulfilled once the voter has cast his/her vote and must be preserved during the counting process.

Integrity. A voting system has to protect the vote against manipulation once it is cast and until it is counted. Therefore the channel must to provide measures to prevent and/or detect any attempt to change the voter's intent once the vote has been cast.

Voter Verifiability – Cast as Intended. According to Neff and Adler [19] we divided voter verifiability as: "cast as intended verification" and "counted as cast verification". In cast as intended, voter must have the possibility to check that his/her vote has been accurately recorded. In the case of remote voting, this implies the availability to check if the vote received by the Election Officials and stored in the remote Ballot Box (in a physical or electronic manner) is the same as cast by the voter. It is important to note that this requirement cannot conflict with the others ones (i.e., coercion and vote buying).

Voter Verifiability – Counted as Cast. In the counted as cast verification, voters must have the possibility to verify the inclusion of his/her vote in the final tally. This is not a requirement currently demanded by traditional voting methods. However we consider it as a security improvement.

Prevention of Intermediate Results. It is important to prevent the disclosure of intermediate results before the election is closed. This way, all the voters have the same information during the voting stage. This implies that the secrecy of the vote must be preserved until the tally process.

Ballot Box Accuracy. Protection of the ballot box against the addition of bogus ballots or the elimination of valid ballots is needed. In the case that multiple voting is allowed, this measure must guarantee that only one vote per voter will be counted.

Coercion and Vote Buying Resistance. As introduced before, one of the main concerns of remote voting channel is that it facilitates coercion or vote buying. Therefore it is important to verify if the channel facilitates these practices

or includes countermeasures to prevent them. The voting channel must mitigate the voluntary impersonation, in which eligible voters cooperates with the coercer or buyer to access the voting system.

Channel Reliability. Most government efforts are focused to increase the reliability of their current remote voting channels. Voters wish to know if their vote has been received by the electoral authority on time to be tallied. As introduced above channel availability is not only related to the delivery speed of the channel. Other factors, as the risks of denial of service attacks, influence the availability of the channel. Therefore in this criterion we will balance the availability to detect such delays in an appropriate timeframe (e.g., the detection of a denial of service) and the availability to react to them (e.g., use a contingency channel to cast the vote).

3.2 Usability

Usability mainly comprises those aspects related to voter convenience while interacting with the voting system. It is important to evaluate if the different channels have important issues or improvements in the usability of the voting process to prevent disenfranchisement.

Prevention of Voting Errors. The voting channel has to prevent involuntary voting errors by voters when casting their votes (e.g., under-voting, over-voting). This practice is becoming more common for poll-site voting in complex elections. In this study we will analyze if the remote voting channel has provisions to prevent such errors.

Ease of Use. The voting channel must be easy to use by average voters. In remote voting this requirement is of paramount importance to prevent disenfranchisement and facilitate the participation of voters.

Accessibility. Disabled voters have to be allowed to vote with total privacy without the need of assistance from third parties. In countries such as Australia, remote voting is one of the main channels used for disabled voters to cast their votes. Therefore it is important to analyze if the channel also provides additional accessibility improvements to such voters.

3.3 Election Management Issues

Finally a remote voting channel must be easy to manage by administrators and electoral authorities. The addition of remote voting does not imply a great increase on the complexity of the election management to guarantee its successful adoption.

Election Set-Up. The voting channel has to be suitable to carry out a single election set-up. This requirement regulates the time to set-up the election and to provide voters with the voting material.

Voting Period Election Management. The voting channel has to be easy to manage during the voting period. This management measures the effort required by election managers to follow-up the participation during the voting process, provide support to voters, and react if any issue arose during the election.

Counting Process. It is important that the voting channel does not delay the current counting process. This criterion evaluates the effort (resources and time) needed to count the votes at the end of the election.

Auditing of the Election Results. Voting channels must provide means for facilitating the audit of the election to ensure its correct execution. This means that it must allow the verification of the election results accuracy and help to solve any dispute.

4 Analysis

After defining the comparison criteria for the different voting schemes, we need to define how each criterion will be evaluated. At this point, we distinguished two different approaches depending of the type of criteria. For security requirements we used a Risk Management approach, which is the most common way for evaluating the security of the system. In the case of usability and election management criteria, we used a basic requirement fulfilment approach.

When implementing the Risk Management of the different voting schemes we started making a risk assessment in which we identified the risks related to each security criteria for each voting scheme; and a risk mitigation analysis in which we evaluated the vulnerability level of each scheme after considering the security controls implemented by them.

During the risk assessment we defined as the main vulnerabilities those that can compromise the security criteria defined. After this, we compiled the threats

Table 1. A short example of risk assessment for two particular attacks

Attack	Channel	Objective	Assumptions	Effort required	Probability of detection	Security controls	Efficiency of security controls	Risk acceptance
Virus	Internet Voting	Vote integrity - modify the voter intent	The attacker does not know which voting terminals are going to be used to cast a vote. Voting terminals interacts with the voting system using an applet.	A virus has to be widespread in order to reach its objective. The virus has to be able to manipulate the applet behaviour	High (a widespread attack is more suitable to detect than a targeted attack)	Antivirus. Network sensors	Medium (antivirus and network sensors cannot control all of the Internet)	Higher than postal voting
Intrusion in the postal delivery channel	Postal Voting	Vote integrity - modify the voter intent	Ballots are transported and stored without strict security measures	Evade watchfulness of ballots	Low (due to inefficient custody)	Physical access control measures	Low (it is practically impossible to protect the delivery channel)	Medium (currently accepted)

against these vulnerabilities for each voting channel. Finally, we described the risks related to each threat, considering, among others: the assumptions required for exploiting the threat, the effort required for exploiting it, the probability of detecting the attack and the availability to isolate the compromised information (e.g., if we can isolate a compromised vote from the rest). Since the risks could be grouped by the vulnerability exploited, we can evaluate the risks for each security criteria by channel.

Once we had defined the risks, we identified the security controls (i.e., countermeasures) that can be applied for each risk to mitigate it. In addition we evaluated the efficiency of this countermeasure on detecting and mitigating the risk. Finally, using the current risk acceptance level on postal voting as refer-

Table 2. Comparative of security requirements

Comparative factor	Postal Voting	Fax Voting	e-mail Voting	Internet Voting
Eligibility	**Medium** Easy impersonation to cast a vote. Handwritten signatures are difficult to validate accurately or not always validated.	**Low** Easy impersonation to cast a vote. Handwritten signatures are digitalized and therefore easy to tamper with.	**Low** Easy impersonation to cast a vote. Handwritten signatures are digitalized and therefore easy to tamper with.	**High** Using strong authentication such as digital certificates, meansonly eligible voters can cast a vote.
Privacy	**Medium** There is a risk that the contents of postal votes could be accessed during their transportation or when the postal envelopes are opened.	**Low** Votes are received without any privacy protection. Voters are required to sign a secrecy waiver.	**Low** Votes are received without any privacy protection. Voters are required to sign a secrecy waiver.	**High** Votes are encrypted before being cast. Cryptographic measures, such as mixing processes, can be implemented to break any connection between vote and voter. Voters can protect their PC's against malware or use secure voting kiosks.
Integrity	**Low** There is no way to prove that the cast vote stays unaltered during the election process.	**Low** There is no way to prove that the cast vote stays unaltered during the election process.	**Low** There is no way to prove that the cast vote stays unaltered during the election process.	**High** Votes can be digitally signed, preventing any manipulation. Furthermore, when using voting receipts, any attempt to delete a vote could be detected by the voter when verifying the receipt.
Voter verifiability - cast as intended	**Medium** There are tools to track a vote sent by mail. However, there is no guarantee that the envelope received by the Election Officials contains the same vote cast by the voter.	**Low** There is no guarantee that the Fax vote is received at the destination as it was cast by the voter.	**Low** There is no guarantee that the postal envelope that contains the vote is received at the destination as it was cast by the voter.	**High** A verification process can be implemented as an independent process from the vote selection process in the voting terminal.
Voter verifiability - counted as cast	**Medium** The voter can verify that his / her ballot is present during the tallying process through a ballot tracker. However, the voter can not verify if the ballot contents are the same selected by him /her.	**Low** The voter does not have any means to individually verify that her cast vote is present during the tallying process.	**Low** The voter does not have any means to individually verify that her cast vote is present during the tallying process.	**High** A voting receipt allows voters to individually verify that their votes are present in the tallying process.

Table 2. (*continued*)

Comparative factor	Postal Voting	Fax Voting	e-mail Voting	Internet Voting
Prevention of intermediate results	**Medium** The contents of the votes and therefore intermediate results could be accessed during transportation.	**Low** Vote contents could be accessed during the transmission. The vote contents are always accessible upon reception.	**Low** Vote contents could be accessed during the transmission. The vote contents are always accessible upon reception.	**High** Votes are encrypted before they are cast. Only the board members can decrypt them at the end of the election.
Ballot box accuracy	**Medium** It is possible to add bogus ballots without detection. Votes can also be eliminated during transportation. Handwritten signatures can be verified to detect massive fraud.	**Medium** It is possible to add bogus ballots without detection However, the fax numbers of the voters can be audited in order to detect mass fraud.	**Low** It is possible to add bogus ballots. Email addresses can be impersonated. Also emails can be eliminated during transmission.	**High** Each encrypted vote can be digitally signed using a unique voter digital certificate to prevent the addition of bogus votes. Additionally, voting receipts can be provided to voters to allow them to detect the elimination of their votes.
Coercion and vote buying resistance	**Low** Voters can show the selected voting options to third parties before casting their votes, making coercion and vote selling possible.	**Low** Voters can show the selected voting options to third parties before casting their votes, making coercion and vote selling possible.	**Low** Voters can show the selected voting options to third parties before casting their votes, making coercion and vote selling possible.	**Medium** If a voter is coerced with the coercer's presence, he/she can cast a new vote later if multiple-voting is allowed. Alternatively, voting kiosks could help to prevent coercion and vote buying.
Channel reliability	**Low** This voting channel depends on the reliability of the postal system of the country from which votes are cast. It is not unusual to receive votes after the closing date and voters can not do anything.	**High** Voters realize if their Fax vote has not reached to the election authority. Therefore contingency measures (e.g., try later or use another voting channel) can be used to prevent the lost of their votes.	**Medium** E-mail reception confirmation can be sent to the voter. However the e-mail transmission can be delayed.	**High** Voters realize if their vote has not reached the election authority if an error arises when casting the vote. Therefore contingency measures (e.g., try later or use another voting channel) can be used to prevent the lost of their votes.

ence, we evaluated the risk acceptance. For example, if we found that the risk of exploiting a postal voting vulnerability is higher than the other channel, we evaluated the criteria as higher than postal voting. Lower and Medium are used in case the risk is worst or the same as postal voting. In the case of postal voting, we used the value of the risk as reference.

Table 1 shows a short example of the risk evaluation method. In this example two different attacks (acting on two different voting channels), both with the objective of modifying the voters intent are compared.

In the case of usability and election management, we ranked each voting channel as Low, Medium or High according to the level of requirement fulfillment.

To summarize the results of the analysis we made a set of tables that compiles the requirements of each criterion. In each case we also remark which are the main factors that contributed to its ranking.

4.1 Security Requirements

The table 2 summarize our findings on the level of fulfilment of the security requirements previously explained.

Table 3. Comparative of usability requirements

Comparative factor	Postal Voting	Fax Voting	e-mail Voting	Internet Voting
Prevention of voting errors	**Low** Voters cannot be alerted of involuntary voting errors that can invalidate their votes.	**Low** Voters cannot be alerted of involuntary voting errors that can invalidate their votes.	**Medium** Involuntary errors cannot be detected when the vote is being cast. However, votes can be automatically reviewed when received to detect errors. This could be a complex process since it requires some email interchanges with the voter (e.g., sending an email response with the vote status to the voter and wait for voter email confirmation).	**High** The voting application can detect any error during the selection process and notify voters before they cast their final vote. This allows voters to make the appropriate corrections.
Ease of use	**High** Most voters are familiar with paper ballots.	**High** Most voters are familiar with paper ballots.	**Low** Voters usually need to scan their votes and attach the scanned image to emails.	**Medium** Voting terminals can provide an intuitive, easy-to-use voter interface with clear instructions. However, some voters are not familiar with this kind of devices.
Accessibility	**Low** Paper ballots possess serious problems to visual impaired and some physically disabled voters. Support of multiple languages can cause privacy issues.	**Low** Paper ballots pose serious problems to visual impaired and some physically disabled voters. Support of multiple languages can cause privacy issues.	**Medium** The use of personal computers facilitates the interaction for disabled voters. However support of casting votes in multiple languages can cause privacy issues.	**High** The use of personal computers facilitates the interaction for disabled voters. Support of multiple languages does not require casting the vote in that language.

Table 4. Comparative of election management requirements

Comparative factor	Postal Voting	Fax Voting	e-mail Voting	Internet Voting
Election set-up	**Low** Setting up an election requires long timeframes to ensure that election materials are received by voters on time.	**Medium** Time frames can be reduced since election materials can be received by fax. However this process could be difficult to automate.	**High** The sending of election materials can be automated by email.	**High** Process can be centralized (e.g., generation of a centralized electoral roll, instructions in a website) without requiring sending any materials to voters.
Voting period election management	**Medium** The management of the postal votes is mostly manual.	**Low** The privacy of the faxed votes must be protected (e.g., introduced in postal envelopes) after being received.	**Medium** The management of the votes could be automated. However email formatting problems could arise (e.g., different scanning resolutions).	**High** The management can be automated. Only the management of the security measures (e.g., management of election cryptographic keys) could add some complexity.
Counting process	**Medium** Votes must be manually counted or Election Officials have to place them into an optical counting device.	**Medium** Votes must be manually counted or Election Officials have to place them in an optical counting device.	**High** Counting process can be automated. However, problems with the internal formatting of votes could arise.	**High** Counting process is automated.
Auditing of the election results	**Low** There is no guarantee that the cast vote is present during the tallying process. The voting channel (postal service) is practically impossible to audit.	**Low** There is no guarantee that the cast vote is present during the tallying process. Voting channel (land phone) is difficult to audit.	**Low** There is no guarantee that the cast vote is present during the tallying process. Voting channel (mailers, DNS servers, etc.) is difficult to audit.	**High** Voters can individually check the accuracy of the election with their voting receipts. Auditors can audit the voting application.

4.2 Usability Requirements

Table 3 summarizes the level of fulfilment of the usability requirements previously explained.

4.3 Election Management Requirements

Finally table 4 shows how the management requirements are fulfilled for each voting channel.

5 Conclusions

There is a general concern that the use of remote electronic voting channels generates more security issues than postal voting. However, after comparing the different channels used in the United States, we have seen that Internet voting concerns have similar implications as the postal voting ones. Furthermore, Internet voting allows the implementation of adequate security measures (such as cryptographic voting schemes) that can reduce the exposure risks of the remote voting methods currently accepted in the United States. That does not mean that the Internet voting channel is not facing any security risks, but it provides a better framework for managing the risks. An example are risks that threaten the reliability of the channel. Internet voting is not exempt from denial of service attacks that could prevent voters from casting their votes. However, these attacks are detected by the voters when casting their votes and can immediately react against them (e.g., cast their votes later or using another voting channel). In the case of postal voting voters are not aware of such attacks (e.g., the capture of postal votes or election material) therefore these attacks could be more effective in such channel. From a vote integrity point of view, Internet voting is susceptible to virus attacks that can change the voters intent. However the exploitation of such risk requires a widespread attack (since the attacker does not know which machines are used for voting) that makes it easier to detect and react against it. In the case of postal voting an attacker knows that votes are delivered without any protection through the postal channel, and can therefore select a target from where to manipulate them (e.g., a central postal office). This attack is more likely to remain undetected than the previous Internet one and therefore the risk is higher.

Regarding fax and email electronic voting channels, they pose severe concerns to the secrecy of the vote. Furthermore the nature of these channels (e.g., the use of unsecure land telephone networks or public email relays) makes it more difficult to implement adequate security countermeasure that mitigate security issues that could threaten the election accuracy. Therefore they cannot be considered a good alternative to postal voting.

In addition we compared the usability and management requirements of the different channels. Again, the use of Internet voting does not pose any additional concerns to the fulfilment of such requirements. Furthermore, it also provides some improvements in both cases. For example, Internet voting prevents the involuntary casting of invalid votes and increases the voting process' accessibility.

Moreover, Internet voting does not increase the complexity of the election process and it improves the election auditability.

References

1. Alvarez, R.M., Hall, T.E., Roberts, B.F.: Military Voting and the Law: Procedural and Technological Solutions to the Ballot Transit Problem. Institute of Public and International Affairs 16, 1–59
2. Republique Et Canton de Geneve. E-Voting. Available at: http://www.geneve.ch/evoting/english/welcome.asp
3. May 2007 Pilot Schemes. U.K (2007), Available at: http://www.electoralcommission.org.uk/elections/pilotsmay2007.cfm
4. Internet Voting in Estonia. Available at: http://www.vvk.ee/engindex.html#0003
5. The Parliament of the Commonwealth of Australia - Joint Standing Committee on Electoral Matters, The 2004 Federal Election - Report of the Inquiry into the Conduct of the 2004 Federal Election and Matters Related Thereto, September 2005, Canberra (Australia) (2005) (ISBN 0 642 78705 0)
6. United States Department of Defense (2007). Expanding the Use of Electronic Voting Technology for UOCAVA Citizens (May 2007)
7. Krimmer, R., Volkamer, M.: Bits or Paper? Comparing Remote Electronic Voting to Postal Voting. In: Wimmer, M.A., Traunmüller, R., Grönlund, Andersen, K.V. (eds.) EGOV 2005. LNCS, vol. 3591, Springer, Heidelberg (2005)
8. Jefferson, D., Rubin, A., Simons, B., Wagner, D.: A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE) (2004), Available at: http://servesecurityreport.org/paper.pdf
9. Voting over the Internet (VOI) Voting Project (2000). Available at: http://www.fvap.gov/services/voi.html
10. Qvortrup, M.: First past the Postman: Voting by Mail in Comparative Perspective. The Political Quarterly 76(3), 414–419 (2005)
11. Uhlmann, C.: Polls Apart. About the House. Australian House of Representative Magazine, (24) (August 2005)
12. Riera, A., Cervello, G. (2004). Electronic Voting in Europe. Technology, Law, Politics and Society, Workshop of the ESF TED Programme together with GI and OCG, July, 7th–9th, 2004, in SchloßHofen / Bregenz, Lake of Constance, 91–100 (2004)
13. VoteHere. Mail-in Ballot Tracker, http://www.votehere.com/ballottrackermailin.php
14. Federal Voting Assistance Program - U.S. Department of Defense, Appendix B Electronic Transmission of Election Materials. Voting Assistance Guide (2006), Available at: http://www.fvap.gov/pubs/vag/pdfvag/appendix-b.pdf
15. California Internet Voting Task Force (2000). Final report. Available at: http://www.ss.ca.gov/executive/ivote/
16. Riera, A., Brown, P., Ortega, J.A. (2003) Advanced Security to Enable Trustworthy Electronic Voting. 3rd European Conference on e-Government. Trinity College, Dublin, (June 2003)
17. Jefferson, D.: Requirements for electronic and internet voting systems in public elections. In: WOTE (2001)
18. Gerck, E. (2001). Internet Voting Requirements. The Bell 1(7), 3 (2000)
19. Neff, A., Adler, J.: Verifiable e-Voting Indisputable Electronic Elections at Polling Places. VoteHere Inc (2003)

Internet-Voting: Opportunity or Threat for Democracy?

Emmanuel Benoist, Bernhard Anrig, and David-Olivier Jaquet-Chiffelle

V.I.P. - Virtual Identity, Privacy and Security
University of Applied Sciences Bern – Quellgasse 21, CH-2501 Biel

Abstract. During the last decade, Internet-voting (i-voting) moved from the field of fundamental research to practical application. First, we will see that theoretical research provides satisfying algorithms for some of the challenges raised by i-voting and that some real world experiments have already been developed and performed.

Unfortunately, in current i-voting systems, the citizen loses his/her control over the overall electoral process. Indeed, only insiders usually have access to the programming code of the application and to the servers used in i-voting. The confidence in democracy itself could be harmed by this opacity.

The European Convention on Human Rights emphasizes that votes should remain secret. This can not be assured for i-voting, since it is not possible to have a booth around each computer for example during the voting process. Family voting cannot be prevented and vote buying could be a major threat for democracy.

Moreover, we can not assume that the voter's computer contains no viruses or Trojan horses. Therefore, it is optimistic to assume that the ballot transfered to the server is the one chosen by the voter.

Finally, we will see that the effect of i-voting on the turnout at polls might remain marginal.

1 Introduction

For many researchers in the area of computer security, i-voting is a very challenging and motivating topic. We observe also a growing interest from administrations as i-voting is expected to increase the number of voters while saving money during the counting of the ballots. We will see that i-voting will soon become a reality, but that it will also imply some change in the citizen's empowerment for the whole voting process.

In most developed countries, the rate of people choosing to vote has dropped drastically over the last decades. For instance in the general election in Switzerland the turnout at the polls has constantly decreased during the last century passing form 80 percents in 1919 to 45 percents in 2003 [1]. The same process is visible in all Western countries where voting is not mandatory. Politicians think that the voting process has to be modernized and that it would have a major impact on the turnover. Since Internet is now used extensively by a large proportion of the population for e-banking, e-tickets for airlines or train, etc. the

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 29–37, 2007.

government should offer the same type of service for voting also. They want to offer a way to vote that is as close as possible to the way people live [2].

The penetration rate of cell phones is far above the one of Internet, that is the reason why the administration of the Canton of Zurich in Switzerland even wants to organize e-voting using cell phones. Such a system has already been tested for the election of students representatives at the University of Zurich [3]. The same system has also been tested on a small scale for real votes (also in the Canton of Zurich).

Moreover, organizing elections costs a lot of money. It is tempting to use new technologies in order to reduce the costs. Private industry uses Information Technologies to increase productivity; why should the administration not use new e-voting technologies, like i-voting, to gain productivity and save money? This argument is particularly relevant in a direct democracy such as Switzerland for example, where voters have to cast a ballot typically four times a year and where each ballot paper contains several questions (local, cantonal and/or federal ones). The traditional handling of the ballots — i.e. counting by hand — is expensive and time-consuming. This explains why some administrations are in favor of i-voting, hoping to save time and money. Last but not least, i-voting carries also the positive image of a modern technology.

2 Encouraging Results

In the last decade, several encouraging results have been found in the area of i-voting. First, researchers from the area of computer science security have developed protocols to ensure a secure voting process. Then some countries (or regions) have developed and introduced i-voting at a larger — but still limited — scale, proving its feasibility: this was for example the case in Geneva, Neuchâtel and Zurich in Switzerland [3,4,5].

Many publications have been written presenting i-voting schemes. Most of them respect the following golden rules of i-voting presented by Cranor and Cytron [6]:

- **Accuracy**
  - A cast vote can not be altered.
  - An invalid vote is not counted.
  - Each voter has the guarantee that his/her ballot is counted.
- **Democracy**
  - Only an authorized voter can participate.
  - Each voter can cast only one vote.
- **Privacy**
  A ballot can not be linked back to the voter who cast it.
- **Verifiability**
  Each voter can verify that his/her vote is counted.

Ray et al. [7] defined an additional rule: **No Unauthorized Proxy**. If a voter decides not to cast his/her ballot, no third party can take advantage of this and cast a forged ballot.

It is possible to design a voting protocol fulfilling all those requirements. Most of the algorithms [8,9,10] use anonymous channels to cast the ballot as an untraceable message. Ray et al. developed an algorithm using very soft cryptography; voting is done in five phases:

1. Blank ballot distribution: the voter identifies himself/herself with his/her certificate and gets a blank ballot.
2. Generating a voter mark: the voter generates a mark from this ballot.
3. Voter certification: the voter gets a blind signature on his/her voter mark.
4. Vote casting: the voter transfers the signed voter mark together with its ballot to the server.
5. Vote counting: once the voting period is over, all cast ballots are made public and the results are announced.

Some countries have already introduced i-voting on a significant scale. For instance, a few Swiss cantons have implemented i-voting schemes and are testing them already in real votes. The main principle (for the Canton of Geneva) is simple. Each voter receives a voting card containing a field that can be scratched off. This field contains a one-time secret password for the voter. The voter connects to a dedicated server, gives the secret number and can enter his/her ballot using a website accessed by the `https` protocol[1]. Actually the i-voting implementations of the three Swiss cantons differ. Some implementations seem to be more mature and secure than others.

The three cantons mentioned above have already organized votes with e-voting where the citizens have participated in real elections or voting. The Canton of Geneva has developed an Internet platform for i-voting. The system is dedicated to i-voting only. Additionally, this canton is already planning to introduce i-voting on a larger scale and a new law i.e. a modification of the constitution, is almost ready and should soon be presented to the cantonal parliament[2]. In the Canton of Neuchâtel, i-voting is part of the general e-government portal "Guichet Unique" (GU). Citizens can not only vote using this system, they can also fill in their tax declaration on-line, follow up the tax invoices and tax payment. Not all citizens have a GU account; among the ones who have, around 54% percent voted using GU. The tests in the Canton of Zurich were using two different e-voting platforms: Internet and SMS. For SMS, the voter does not send his/her choice in clear text form (for instance "Yes" or "No"). He or she receives a table containing a code for each possible choice, then the voter sends the code corresponding to his/her choice to the SMS gateway. The votes are anonymized afterward and transfered to an electronic voting box. This test has also been a success; it has been used for the election of the board of students at the University of Zurich. During this election, only one student did vote using a paper ballot, all others used e-voting, i.e. 1582 using Internet and 205 using SMS. Even for a real election (Bülach April 2006), 20.67% of the voters used e-voting, among which 3.85% used SMS voting [11].

[1] `https` is a combination of a normal `http` over a Secure Sockets Layer SSL or Transport Layer Security TLS.

[2] Proposition de loi 9931 et proposition de loi constitutionelle 10013.

It is interesting to point out the significant difference between the complexity of academic crypto-solutions, that are proved to be secure under certain assumptions, and the simplicity of the techniques chosen in current real life implementations. But we have to keep in mind that the requirements are not the same for both cases. In academic research, i-voting is a goal in itself and both security and trustworthiness are top priorities. In real life, one has to try and fulfill new practical constraints that might be incompatible with some secure academic solutions. Indeed, i-voting is currently introduced to make the voting process easier and more attractive, in order to gain new voters. Therefore,

- it should **be very simple and easy to use**: i-voting is intended to be used by many voters; even older or disabled people should have the opportunity to use it,
- it should **not require any special installation**: it should be based on open protocols (`http`/`https`/`ftp`), and should not require any specific client software,
- it should **be compatible with other off-line voting mechanism**, since we probably won't and cannot force all citizens to do i-voting.

These three added constraints explain partly the gap between theoretical results and current practical solutions.

3 Empowering the Citizens?

One of the goals of politicians supporting i-voting is to increase the number of voting persons, and hence improve the democracy. Yet the outcome might be paradoxical, since in many points, introducing i-voting could reduce the power of the citizen and its confidence in the voting process itself and consequently harm the democracy.

In traditional voting processes, citizen usually can verify, either directly or indirectly, the way their ballot has been taken into account.[3] They typically put their ballot in a special box, and can assist later to the counting of the ballots. In the countries where citizen cannot assist, it is possible to have people monitoring this process. Everybody can convince himself/herself that each vote has been counted correctly.

This type of confidence in the system should be saved for i-voting. Administrations have to provide at least the same level of confidence. Some algorithms, such as the one of Ray et al. [7] provide a way for the voter to check if his/her vote has been taken into account. But, the voter must also be convinced that the program used for the i-voting is really working as it is supposed to be. That means, in particular, that it fulfills all the requirements seen above. Unfortunately, in the domain of i-voting, security by obscurity is often the rule. A "normal" citizen can neither access the code, nor see the type of algorithm used, nor check that

[3] There are famous exceptions like the "Landsgemeinde" in Switzerland where votes are expressed by raising one's hand and are visually counted by trusted experts.

the machine is well configured and that the administration or other third parties do not manipulate votes. In traditional voting processes, forging 100 ballots is nearly 100 times more difficult than forging one ballot. This difference does not exist in i-voting, since the fraud can be reproduced almost without extra costs. Manipulating ballots at a large scale seems to be easier and less detectable in i-voting than in traditional voting.

In order to give citizens more confidence in the system, we need a totally transparent i-voting process letting the citizen monitor each of the steps. Confidence in the democratic process can only be reached when i-voting uses transparent boxes. That means that any citizen should be able to access the algorithm, and read the code. In particular, only Open-Source projects should be accepted for i-voting. But this would not be sufficient. Citizens must be able to check if the program they inspected is really the one that is running on the server. There is unfortunately no way to achieve this goal. One can nevertheless try to certify what happens on a server. Trusted computer can for instance certify that the boot-process used by the server was right. It is also possible to use a micro kernel that can be totally verified on such a trusted computer. Unfortunately, such systems are not secure enough and can not be used in a productive environment.

4 Can Voting Remain Secret?

In a traditional voting process, voters are hidden in polling booths while filling in their ballot. There is no such booth in current i-voting systems. Family voting cannot be prevented and vote buying could become a major threat for democracy.

The article 3 of the Protocol 1 of the European Convention on Human Rights says: [12]. This protocol has been signed and ratified by almost all the European countries (except Switzerland). It means that ballots have to be secret; nobody should be able to know who voted for which candidate. With i-voting, one can verify the way a voter votes by forcing him/her to vote under surveillance. Or someone can get the empty voting cards of all his/her clients and can vote for all of them. This can be used for vote buying. There are ways to circumvent this by adding a secret key. If the key is false when voting, then this vote is not counted but both the user and the observer do not see it. However, i-voting becomes more complicated and the risk of an error in such a manipulation is high. The voter cannot see that his/her vote is not valid, he/she might lose his/her possibility to cast a ballot for this election.

The second risk is family voting. The Venice commission of the council of Europe has produced a Code of Good Practice in Electoral Matters [13]. In its article 4 about secret suffrage, it says that . It is also said, that

. That is the reason why they express huge concerns about postal voting. For the council of Europe, it should not be widely encouraged and should be reserved for special cases.

These restrictions should also be extended to i-voting, since it is not possible to physically ensure the secrecy of the votes. There is no way to check if a voter voted free or under control of another family member.

Moreover, even when the transmission over the Internet might be well protected, there is almost no protection of the voters' computers. It is not reasonable to expect for a "normal" user to protect its screen with a Faraday cage in order to not be spied on . . . It it not reasonable to expect that computers are virus free, or that they do not contain any Trojan horse. A fundamental security weakness of i-voting is on the side of the voters which use their own machines.

Rubin [14] presented a study of the security weaknesses for i-voting. One can start with the server that could be controlled by an attacker. The network can also be attacked (Distributed Denial of Service attack for instance). But the major risks are situated in the voter's computer. Some possible attacks may consist in taking the control of the computer, modifying the browser or the operating system, in such a way that the vote transmitted by the computer is different than the vote the voter has chosen. The attacker would manipulate Input and Output so that the voter would not see any problem and would have the feeling everything was right. It is also possible to use a key logger, in order to monitor the way people vote. Since operating systems and programs used on the server and on the client contain typically many thousand lines of code, failures are contained therein. It is not possible to formally verify each piece of code [15,16]. Those failures can (and will) be used by hackers to access the system. Security measures can be taken, but zero risk is impossible to be reached nowadays. Since attackers may be states (a foreign state or even the state organizing the vote), there is potentially no limit to the power of the attackers.

In order to control the result of a tight vote, attackers don't need to take control of all the computers of the voters. For a tight election, say 51% to 49%, controlling only 3% of the votes would allow to change the result of the election. Since the 3% are unknown, only half is changed from the former winner to the former loser (the other half of the votes are already destinated for the former loser) so 1.5% is transfered, which makes the result changing from 49.5% to 50.5%.

5 Is It Worth the Price?

One of the goals for governments to use i-voting is to increase the proportion of people that are voting. It is difficult to assess if i-voting is a long term solution to motivate people to vote. However, we can already study the effects of the generalization of postal voting on the turnout in Switzerland. Almost all Swiss Cantons have generalized the postal voting in the middle of the 90's (and already

the late 70's for some cantons). In these cantons, any citizen receives one ballot that can either be used for traditional voting or postal voting. We can see in Fig. 1 that the number of voting citizens has not significantly increased since the middle of the 90's. At least it stopped decreasing [17,1]. Two studies have been conducted in Switzerland: a team of the University of Geneva (c2d – centre d'étude et de documentation de la démocratie directe) estimates the potential of gain to be 9% whereas study of Hanspeter Kriesi from the University of Zurich estimates the gain to be at most 1.7%. [11]

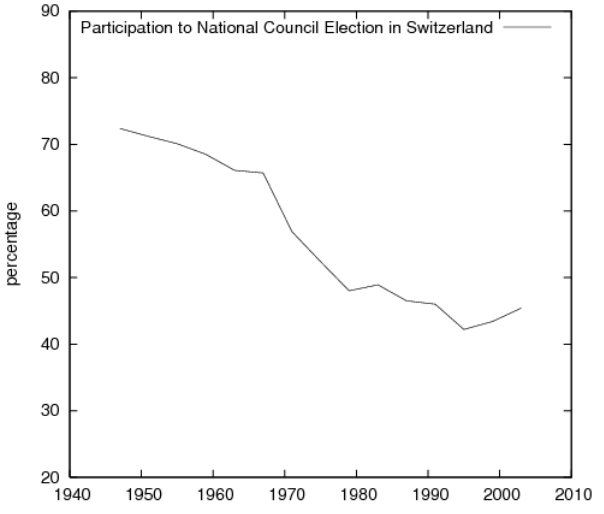

Fig. 1. Voters for the General Elections in Switzerland (1995: generalization of postal voting)

As we see, the effect of the generalization of postal voting in Switzerland has been positive, but limited. Nevertheless, in Switzerland, it is widely accepted that the added value of postal voting, in terms of convenience, is worth the price in terms of security: making easier family voting.

However, it is questionable to assume that the added value of i-voting, in term of convenience and in comparison to postal voting, is worth the extra risks (vote buying, large scale frauds, loss of the citizens' confidence in a fundamental democratic process, etc.) brought by current implementations. In countries where postal voting is not implemented, we could expect the introduction of i-voting to have an impact similar to postal voting in Switzerland as it brings a similar added value in terms of convenience. As we see, this impact — though positive — might remain marginal.

Current i-voting experiments (mis)use the confidence that people have in traditional democratic processes with a long successful history. Any abuse of this confidence might be very counterproductive in the long run. What will the impact on the proportion of voters be, when citizens realize that they have lost most of their control on the voting process? Citizens may even lose confidence in

the whole democratic system, since they can not be sure about the way people are elected or decisions are taken.

We recommend that governments and administrations be very careful with the introduction of i-voting and apply the precautionary principle. If an i-voting system can fulfill all the requirements presented in this paper (securely proven algorithms, simple implementation, open-source program, and transparent computer) and its use is limited to absentee voting (hospital patients, persons in custody, persons with restricted mobility and electors resident abroad to vote), then i-voting could become a successful application of a new technology. As for today, there is no way to have a transparent server, since clients are still totally insecure, it is premature to introduce i-voting and administrations should wait before deploying such a system. Otherwise, it might as well reveal itself as major threat for democracy.

References

1. Swiss Federal Parliament (National Concil). Turnout at the polls from 1919 to 2003, http://www.parlament.ch/e/homepage/wa-statistiken-diagramme/wa-statistiken-diagramme-wahlbeteiligung-ab-1919.htm
2. Hensler, R.: Les nouvelles technologies, ferment de la modernisation de l'etat. In: 5e symposium suisse eGovernement, Swissôtel Zurich-Oerlikon (2004), http://www.geneve.ch/evoting/discours_20040826.asp
3. Statistiches Amt des Kantons Zürich. Abgeschlossenes Pilotprojekt e-Voting des Kantons Zürich, http://www.statistik.zh.ch/produkte/evoting/index.php?p=5
4. Canton de Genève: Site de l'état de Genève consacré au vote par internet, http://www.geneve.ch/evoting/
5. Canton et République de Neuchâtel. E-democratie, http://neuchatel.ne.ch/profils/politique.asp/1-11-160-12345-5001-1001-1-1-2-1/2-0-2345-5001-1000-2-0/
6. Cranor, L.F., Cytron, R.K.: Design and implementation of a practical security-conscious electronic polling system. Technical report (March 25, 1996)
7. Ray, I., Ray, I., Narasimhamurthi, N.: An anonymous electronic voting protocol for voting over the internet (November 26, 2001)
8. Boyd, C.: A new multiple key cipher and an improved voting scheme. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 617–625. Springer, Heidelberg (1990)
9. Chaum, D.: Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, Springer, Heidelberg (1988)
10. Juang, Lei: A secure and practical electronic voting scheme for real world environments. TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems (1997)
11. Swiss Federal Chancellery. Rapport sur les projets pilotes en matière de vote électronique (May 2006), http://www.admin.ch/ch/f/ff/2006/5205.pdf
12. European Court of Human Rights. European convention on human rights and additional protocols. Registry of European Court of Human Rights (September 2003), http://www.echr.coe.int/echr/

13. Council of Europe - Venice Commission. Code of good practice in electoral matters, p. 118 (July 2002), http://www.ohchr.org/english/law/compilation_democracy/docs/compil_demo cracy.pdf
14. Rubin, A.D.: Security considerations for remote electronic voting. Commun. ACM 45(12), 39–44 (2002)
15. Hoglund, G., McGraw, G.: Exploiting software: how to break code, p. 471. Addison Wesley, Reading (2004)
16. McGraw, G.: Software security: building security in. Addison-Wesley software security series. Addison Wesley, Reading (2006)
17. Swiss Federal Chancellery. Enquête sur le vote par correspondance (1999), http://www.admin.ch/ch/f/pore/va/doku/pdf/enquete_bsa.pdf

Assessing Procedural Risks and Threats in e-Voting: Challenges and an Approach

Komminist Weldemariam[1,2], Adolfo Villafiorita[1], and Andrea Mattioli[1]

[1] Fondazione Bruno Kessler (FBK-IRST)
{sisai,adolfo,amattiol}@fbk.eu
[2] ICT International Doctorate School
University of Trento
weldemar@dit.unit.it

Abstract. Performing a good security analysis on the design of a system is an essential step in order to guarantee a reasonable level of protection. However, different attacks and threats may be carried out depending on the operational environment in which the system is used, i.e. the procedures that define how to operate the systems. We are interested in reasoning about the security of e-Voting procedures, namely on the risks and attacks that can be carried out during an election. Our focus is more on people and organizations than on systems and technologies. In this paper we describe some ongoing work that we are carrying out within the ProVotE project (a project sponsored by the Autonomous Province of Trento to switch to e-Voting for local elections) to analyze and (possibly) improve procedural security of electronic elections. To do so, we are providing models of the Italian electoral laws using the UML and we are developing a custom methodology for analyzing threats from the models. Our reasoning approach is based on asset mobility, asset values and existence of multiple instances.

Keywords: Procedural Security, Modeling, Threat Actions, e-Voting.

1 Introduction

e-Voting is the subject of great debates and its adoption in various countries has been slow and/or cause of controversies. (see, e.g., [1,2]). A major concern is security: without proper protection and effective control procedures, malicious actors may instantiate a range of threat actions, with effects varying from a "denial of service" (e.g. stopping the election in a polling station by sabotaging some e-Voting machines) up to alteration of the results (e.g. by successfully changing votes in some key precincts).

Interestingly, paper voting and the procedures regulating "paper" elections have weaknesses and possible attacks, that can usually be carried out under the hypothesis of multiple "failures" (e.g. a ballot is stolen before the election and the polling officers do not realize it). The usage of electronic devices, however, shifts and amplifies some of the risks.

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 38–49, 2007.

We are interested in reasoning about the procedural security of elections, and, in particular, electronic elections, that is, on the procedures and controls that regulate the usage of e-Voting machines. We focus, in particular, on techniques to highlight possible threats, in order to derive security and assurance requirements.

Some interesting approaches to perform procedural security have been proposed in the past in [3,4,5,6], and other general security analysis techniques in [7,8]. Voting, however, differs from the environment in which these methodologies are most effective for the following reasons (see also [2,9,10]):

- : assets and sensible data related to an election are handled (and may be altered) by different actors (e.g., technicians, poll officers, electoral officers) with different responsibilities over time and in different locations;
- : assets related to an election change their value over time. The effects of an attack on an asset change dramatically according to the period in which the attack is performed (e.g. tampering with an e-Voting machine **after** the election does not have much of an effect);
- : various electoral assets need to be replicated for running an election. The effects of an attack may not only propagate to copies, if the master is compromised, but may also have a different impact, depending on the number of instances that are affected by the attack (e.g. breaking one e-Voting machine may not have a tangible effect on an election).

In order to deal with the points mentioned above, we devised a methodology for modeling (electoral) processes and for modeling and analyzing threats. The methodology is based on the following steps.

- First, we provide a model in the UML that describes the procedures under analysis. The model, mainly based on activity diagrams, provides information about the assets handled by the procedures and their relevant states, called asset-states (e.g. responsibility, location, value, number of instances). The model must follow precise conventions, that we defined in the form of stereotypes that can be used (we defined some stereotypes to enrich the meaning of certain UML elements), what UML elements and constructs can be used, how the model has to be structured, the minimum set of information that has to be provided for each procedure and asset.
- Secondly, we provide a set of threat actions that can be performed on the assets. Threat actions modify the values and/or the state of assets. We group threat actions into simple actions (such as delete, destroy, use, etc) and composed actions (such as replace, copy, etc).
- Thirdly, the threat actions are in the model in order to generate an . In the extended model attacks can be performed, namely threat actions may be composed and executed and cause changes of states of assets.
- Finally, we analyse the extended model in order to evaluate the impact of attacks. The asset-domain model combined with the asset-threat model is currently analyzed by hand. However, the problem can be expressed in terms

of a satisfiability problem and thus allow the usage of tools such as model checkers [11].

The methodology described above is being applied within the ProVotE project to electoral laws of the Autonomous Province of Trento and of Regione Friuli-Venezia Giulia, two Italian territories that have autonomy over local elections and are considering a transition to e-Voting. The ProVotE project, sponsored by the Autonomous Province of Trento, has the goal of introducing e-Voting for local elections. Within the project we are defining the law, the procedures, and the e-Voting system that will be used for local elections. So far, the ProVotE e-Voting system has been used in five different trials involving about 11,000 citizens (see [12,13] for some more details about the project).

This paper describes the work we are carrying out concerning the definition of the procedures and shows some of its advantages and limitations of the approach we propose. In particular, we provided models that describes the electoral procedures and we are now using such models to investigate threats. The final goal is that of providing a set of changes to the local laws that regulate e-Voting.

The rest of the paper is structured as follows. The next section provides some hints on the modeling methodology we have defined. Section 3 describes how we characterize assets, an important detail to perform threat-analysis. Section 4 describes how we perform threat-analysis. Finally, Section 5 and Section 6 provide some comparison to related work and some conclusions.

2 A Methodology for Modeling Electoral Process

The introduction of new technologies in the polling stations changes the risks and attacks that can compromise or invalidate an election. Such risks and attacks not only depend upon the security level the new systems provide, but also on the procedures and controls regulating the way in which the systems are operated.

In order to understand the legal, technical, and administrative issues related to the introduction of e-voting systems, therefore, we believe that a proper (business) process modeling is essential.

As a minimum requirement, such process model should at least be:

1. [illegible] all the stakeholders and, in particular, stakeholders with little or no technical background, should be able to read and understand the model. This is essential in order, e.g., to disambiguate the interpretation of certain norms, by using the model as a common language between (business) analysts and people with legal background;
2. [illegible] all the significant elements in the law should be mappable to elements of the model (e.g. actor and responsibilities, resources and operations on them, constraints);
3. [illegible] all the items in the model should be given a precise and unambiguous meaning, in order to enable simulations and/or of formal verification;
4. [illegible] the organization of the model and of its elements should follow precise and machine-verifiable conventions, in order to guarantee a minimum level of quality, uniformity, and maintainability.

Various works describe how to use the UML for modeling business process (see e.g. [14,15]). However, little is usually said on the third and fourth points above, namely formalization of the model and maintenance. Moreover, some of the peculiarities of the e-Voting domain needed specific strategies for an effective modeling.

We decided therefore to define a tool-supported modeling methodology which could help us address the points above and some of the peculiarities of the electoral domain. The notation basically conforms to the approaches proposed in the past. The "added value" is a set of rules and conventions, described in [16,17], that simplify maintenance and allow to more closely match the notation to some peculiarities of the electoral laws.

The methodology is based on the following elements:

- all the processes (represented with use cases) are organized in a hierarchy. While the higher levels in the hierarchy represent "containers", the leaves represent "executable" processes; the hierarchy is implemented through stereotyped relationships among use cases. The stereotypes allow for "aliasing" (use different names/reuse the same process in various contexts) and for process decomposition. Hierarchical organization, in our experience, helped in browsing and maintaining the model;
- for each "executable" process we specify what actors participate in the process and who is responsible. This information not only allows to describe who does what during the execution of a process, but, more importantly in the context of this work, who manages what data and with what privileges (see also next item). During the execution of the process, the responsible actor is liable for the assets on which the process operates;
- for each "executable" process we provide its specification using an activity diagram. Activities within the diagram are linked to assets they manipulate, using the CRUD notation (create, read, update, and delete). To do so, we use a facility introduced with UML 1.4, that allows to associate activities and entities, and stereotypes to provide the precise meaning of the association. Special stereotypes are used for "passing" assets among processes. The notation allows to determine precisely all the operations that are performed on assets during the execution of the processes.
- for each process we provide traceability to the norms (using a hierarchy of packages and stereotyped classes) and other information (such as the time frame within which the process has to take place and the multiplicity of assets). "Executable" processes, moreover, are documented in natural language for documentation purposes.

The methodology is supported by an extended version of Visual-Paradigm for UML (VP for short) [18]. In particular, the management of model elements and diagrams is performed using the standard facilities VP provides. We have developed a set of plug-ins (using the Java API VP provides to access and manipulate models) for:

1. checking compliance of a model with the rules prescribed by the methodology (e.g. all the correct stereotypes are used, all the required information is provided);
2. exporting the information in human readable form (e.g. PDF, HTML documents);
3. extracting significant information from the model, such as the CRUD matrix (what process does what operations of which assets) and the Actor-Asset matrix (what actor has access to what asset and with what privileges).

The methodology has been applied to the electoral laws of the Autonomous Province of Trento that regulate elections of majors in cities and provincial elections. The model includes three different systems, as the law for electing the major distinguishes among "small" towns (with less than 3000 voters) and "bigger" towns. The models are being used as the basis for the definition of the procedures that will regulate electronic elections.

A second case study is represented by the law regulating local polls in Regione Friuli-Venezia Giulia, for which we defined the model regulating the electronic poll which will be held in November 2007 in two small municipalities of the Region. To say a few things about the model dimensions — i.e., how the model is complex — it contains about 100 processes, 50 actors, 76 use cases, and 49 activity diagrams involving many assets.

3 Asset Properties: A Framework to Understand an Asset

An important aspect of the methodology is the how we characterize assets, which contain all the sensitive information. The features of an asset and the way in which such features are changed by the procedures, in fact, provide a lot of information to understand, e.g., possible weaknesses of the procedures and the impact of attacks. There are various definition given to an asset, however, we follow the following definition given in [19]:

In our approach, assets are characterized by , , , , and (see also Figure 1).

The of an asset determines how certain properties are inherited. The nature of an asset is immutable: it cannot be changed during the execution of the procedures. In our formalization, assets can be (such as names, symbols, keywords, passwords, electronic ballots and electronic data in general) or , when they can contain other asset(s) (e.g. a memory support that can contain electronic ballots).

The of an asset (which we represent qualitatively e.g. ,), allows to reason about the impact of threats (e.g. an attack to an asset with does not cause any harm). The initial value of assets is assigned in the model by the analyst (the execution of an activity may change the

value of an asset). The value of [illegible] is determined by their intrinsic value (determined by the analyst) together with the value of the assets they contain.

Asset might be placed in several locations as well as being in transition between various locations. In our framework, therefore, assets are situated in a [illegible] (e.g. an electoral office, a safe, a [illegible] asset). Breaking into a location or being able to access a [illegible] is a mean to lead an attack against a (contained) asset (e.g. stealing a memory support containing electronic ballots allows to attack the electronic ballots). The initial locations are determined by the analyst; locations can be changed by the execution of an activity; the location of primitive assets corresponds to the location of the containers in which they reside. A container asset can be a relative location for another container assets (see Figure 1).

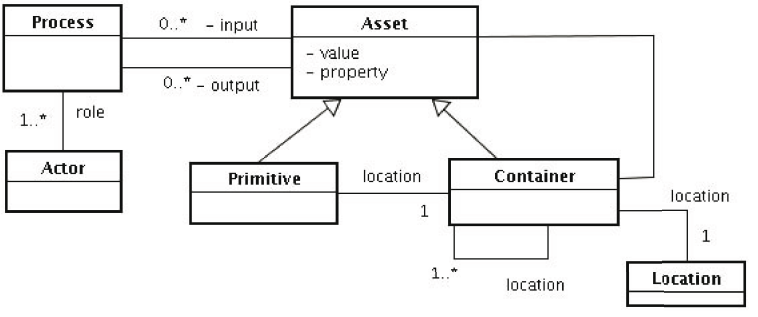

Fig. 1. Characterizing assets

The number of copies (called [illegible]) is particularly relevant in the electoral domain because the same assets are replicated many times, for example, when printing the ballots, and copies go through different responsible people. In addition, this is also important to estimate the effects of realizing a threat against an asset before or afterward a cloning operation.

In addition to the the above mentioned qualities, we also allow to characterize an asset by means of [illegible], that describe the current situation of the asset, such as the security measures that are enacted (e.g. a safe can be [illegible], a file can be stored [illegible] or [illegible]). While for any domain there can be domain-specific states of the asset, we are particularly interested in security states — such as, [illegible].

We call [illegible] of an asset the values of the features of an asset (i.e. value, location, number of instances, and property) at a given instance, and we call [illegible] the sequence of states through which an asset goes during the execution of a process.

Figure 2 shows an excerpt of the procedure that is followed during project trials for delivering the voting software to the polling stations. (The ProVotE e-Voting solution is based on DRE with voters verified audit trail which is installed and used in polling offices; the software for running the machines and

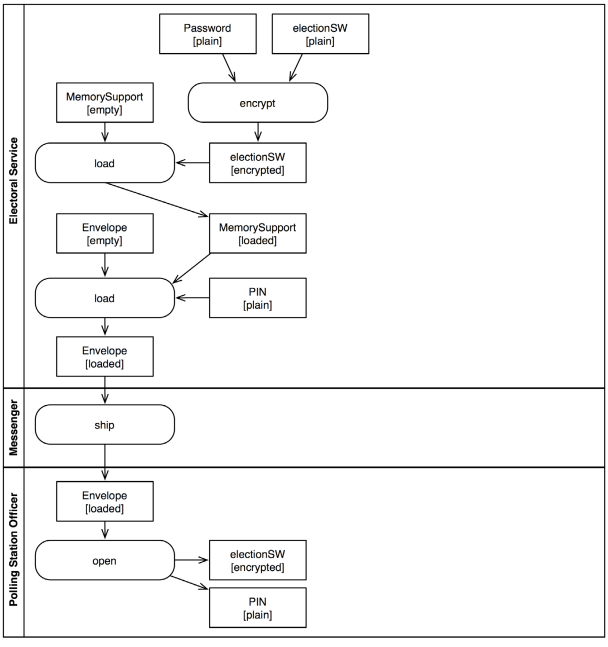

Fig. 2. An example of asset flows

cryptographic keys are prepared and sent to the polling station analogously to what happen for "paper" elections in Italy). The diagram is a simplification both of the assets and of the procedure, for the sake of presentation: various details related, e.g., to the PIN management and to the password used to encrypt/decrypt the software are omitted.

The diagram shows how, before the election, the Electoral Office encrypts the e-Voting software and creates a memory support which contains the final software release. The responsible person at the Electoral Office then prepares an envelope with the PIN code (to activate the voting functions) and the memory support. A carrier (e.g. a police officer, messenger in the diagram) takes the envelope and delivers it to the polling station, where the polling officers, once verified that the envelope is sealed, open it, insert the memory support in the voting machines, insert the PIN and start the voting operations.

4 Asset Threats and Attacks: Understanding the "how"

By assigning each asset a value and a location we can highlight in our model where a threat may be implemented and how much harm it can cause. E.g. a denial of service could be caused by deleting a valuable asset (an asset whose value is non-null) when the asset is in a given location. The notation, however, does not allow to represent the "how", e.g. the flow of actions causing termination

of the procedures in some undesired state, nor to reason about composition of threats, that is, what happens if multiple assets are attacked simultaneously. To tackle these issue, we propose the concepts of [illegible] and [illegible].

We define an **asset threat** as what an adversary might try to do to an asset. More formally we represent threat actions as activities that can transform assets and/or their state. To carry out an asset threat, the adversary may need to execute one or more (more elementary) asset threats against other asset (for example, "adversary reads [password] and signs [data]"). We distinguish therefore between [illegible] and [illegible]. The former are "elementary" actions, while the latter are obtained by composing basic actions to produce more complex behaviors. Figure 3 presents two examples of threat actions, delete and replace. Delete (left hand side of the figure) takes as input an asset and resets the number of instances and the value. Replace (right hand side of the figure) takes as input two assets and returns the second asset. Replace action is a composed action of delete and write: that is, after deleting from an asset some (or all) of its value, the resulting malicious asset is again modified with wrong data and reintroduced in the asset-flow.

Finally, we define an **attack** as a sequence of asset-threats that lead to an undesired state.

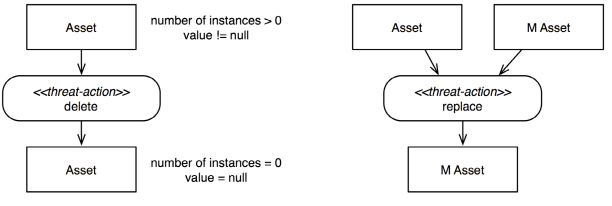

Fig. 3. Two threat actions

In order to analyze what are the possible attacks of a given (set of) procedures, we [illegible] asset threats in the model describing the standard procedures (e.g. what is prescribed by the law) and generate what we call an [illegible]. Thus, in the **extended model** assets are not only manipulated according to what should happen in the nominal case (i.e. according to the electoral laws), but can also be transformed by the execution of one or more asset threat. The **injection** is performed by inserting threat-actions in the domain asset flow.

Figure 4 shows an extended model resulting from the injection of some delete and replace threat-actions in the example of Figure 2 (see discussion below for the injection strategies). The threat actions are marked with the "threat-action" stereotype and are shown in the diagram in red.

The extended model can now be used to analyze the possible attacks. This is done by "executing" the model and analyzing execution paths leading to undesired states. For instance, in the extended model of Figure 4 it is possible to implement at least three different attacks.

Fig. 4. An example of extended model

The first one consists in replacing the software which is sent to the polling stations. By reading the `password`[1] with which the `electionSW` is encrypted and substituting a modified version of the software in the `MemorySupport` it is possible, for a malicious actor, to deliver a modified copy of the software to the polling station. The second one consists in replacing the `PIN`. A malicious actor with access to the `PIN` code may substitute the `PIN` which is loaded in the `envelope` and thus, have a wrong `PIN` delivered to the polling station (eventually causing a denial of service — namely the voting functions cannot be activated by the polling officers). The third attack consists in deleting (or destroying) the `envelope` during transportation, possibly causing another denial of service.

The example, although trivial, shows how — by reasoning on the extended model — it is possible to explicitly represent the attacks that can be carried out, determine what assets are needed by the attackers and when, and who can carry the attacks. Notice that, similarly to what happens in model checking, we do not provide any information about the likelihood of the attacks. However, even in this simple case, we believe that the output the attacks can provide experts

[1] We use `typewriter text` to make reference to entities of the extended model.

the information and the requirements to enhance the current procedures, to eliminate certain attacks or, at least, to make them more difficult to implement.

A few words need to be spent on injection strategies and automation of the analysis, currently carried out by hand. Since attacks depend on what threat-actions are carried out, the effectiveness of the analysis depends upon the injection strategy that is chosen. It turns out that the best injection strategy consists in injecting all possible threat-actions at all possible steps of the nominal procedures. Using this approach analyzing the attacks becomes similar to a model checking problem in which the required final state of some key assets is expressed using LTL/CTL and the counter-example generated by executing the [illegible] contains the sequence of threat-actions causing the final state not to be reached. The approach (injection and analysis) is similar to the one proposed for the FSAP/NuSMV-SA platform for safety analysis [20].

5 Related Work

A proposal for responsibilities and privileges of the actors involved in e-Voting is presented in [21]. The work focuses on Internet voting and has the goal of providing a description of the role of each actor together with the clear indication of what each actor is expected to do with the system processes and defining an operational framework that harmonizes the current (technological) security features of the system and allows to talk about "secure e-Voting system". Differently from this work, our work specifically proposes a modeling in terms of assets and processes and explicitly provides means for assessing threat actions and attacks.

In [3,4], the authors discuss the need for procedural security in electronic elections and provide various examples of procedural risks occurred during trials in the UK. The same authors in [22] also investigate the need for applying business process re-engineering for electoral process. Our work, while sharing the same motivations for procedural security, focuses more on the technical aspects rather than on the motivational aspects, by presenting a methodology to model electoral processes and a technique to represent attacks.

In [6,23], an approach to measure relative security attack surfaces is explained. The authors defined three abstract dimensions for system's attack surfaces: targets and enablers, channels and protocols, and access rights, in order to measure the "attackability" of a system and highlight correlation of bug counts with system vulnerabilities. Moreover, the authors use a state machine model to represent the behavior of an adversary attacking the system. However, differently from our work, their goal is on system design and to determine whether one version of a system is more secure than another with respect to a fixed set of dimensions.

6 Conclusions and Future Work

Analyzing the security of procedures in e-Voting is an essential task to guarantee adequate security levels, as threats and attacks may not only derive from pitfalls in the electronic systems, but also from ill-designed procedures. The transition

to a new way of voting is challenging in several areas, social, technological, procedural and the switch to a new system often has procedural implications and risks that are challenging to assess and evaluate.

Since asset mobility, asset state, asset evolution, and the context in which asset instances are used in e-Voting are an inherent challenge, we have developed a methodology for procedural security analysis which focused on asset and asset flows. The methodology can be used to analyze and evaluate the impact of threats and, consequently, to come out with a set of (security) procedural requirements that guarantee the desired level of protection. We presented the approach by introducing the guidelines we follow for modeling the electoral procedures and hinted its usage through a simple example. Moreover, this kind of reasoning has the potential to serve as a trust building measure in the new e-Voting processes.

The work described in this paper is ongoing. Future work will address finer grained points related to the semantics of asset-flows and automation of the model extension and model analysis via model checking.

References

1. Bryans, J W., Littlewood, B., Ryan, P.Y.A., Strigini, L.: E-voting: Dependability Requirements and Design for Dependability, pp. 988–995. IEEE Computer Society Press, Washington, DC, USA (2006)
2. Rebecca, T., Mercuri, R.T., Camp, L.J.: The Code of Elections. Commun. ACM 47(10), 52–57 (2004)
3. Xenakis, A., Macintosh, A.: Procedural security analysis of electronic voting. In: Rauterberg, M. (ed.) ICEC 2004. LNCS, vol. 3166, pp. 541–546. Springer, Heidelberg (2004)
4. Xenakis, A., Macintosh, A.: Procedural Security and Social Acceptance in E-Voting. In: HICSS 2005: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005) - Track 5, p. 118.1. IEEE Computer Society, Washington, DC, USA (2005)
5. Vetterling, M., Wimmel, G., Wisspeintner, A.: A Graphical Approach to Risk Identification, Motivated by Empirical Investigations. LNCS, pp. 574–588 (Thursday, November 23, 2006)
6. Manadhata, P., Wing, J., Flynn, M., McQueen, M.: Measuring the attack surfaces of two FTP daemons. In: QoP 2006: Proceedings of the 2nd ACM workshop on Quality of protection, pp. 3–10. ACM Press, New York (2006)
7. Braynov, S., Jadiwala, M.: Representation and Analysis of Coordinated Attacks. In: FMSE 2003: Proceedings of the 2003 ACM workshop on Formal methods in security engineering, pp. 43–51. ACM Press, New York (2003)
8. Fovino, I.N., Masera, M.: Through the Description of Attacks: A Multidimensional View. In: Górski, J. (ed.) SAFECOMP 2006. LNCS, vol. 4166, pp. 15–28. Springer, Heidelberg (2006)
9. Krimmer, R., Volkamer, M.: Observing Threats to Voter's Anonymity: Election Observation of Electronic Voting. Working Paper Series on Electronic Voting and Participation Nr. 01/2006 (September 2006). Can be, downloaded from http://static.twoday.net/evoting/files/Working-Paper-1-2006.pdf

10. Prosser, A., Kofler, R., Krimmer, R., Unger, M.K.: Security Assets in E-Voting. In: Prosser, A., Krimmer, R. (eds.) [24], pp. 171–180
11. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Cambridge (2000)
12. Villafiorita, A., Fasanelli, G.: Transitioning to e-Voting: the ProVotE Project and the Trentino's Experience. In: Wimmer, M.A., Scholl, H.J., Grönlund, Andersen, K.V. (eds.) EGOV 2006. LNCS, vol. 4084, Springer, Heidelberg (2006)
13. McGaley, M., McCarthy, J.: Transparency and e-Voting: Democratic vs. commercial interests. In: The International Workshop on Electronic Voting in Europe (2004)
14. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Wohed, P.: On the suitability of UML 2.0 activity diagrams for business process modelling. In: APCCM 2006: Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling, pp. 95–104. Australian Computer Society, Inc, Darlinghurst, Australia (2006)
15. Castela, N., Tribolet, J.M., Silva, A., Guerra, A.: Business Process Modeling with UML. In: ICEIS (2), pp. 679–685 (2001), citeseer.ist.psu.edu/article/castela01business.html
16. Mattioli, A.: Analisi dei Processi in Ambito di Voto Elettronico per le Elezioni in Provincia di Trento. Master's thesis, University of Trento (2005–2006)
17. Manica, A.: Dai processi ai sistemi informativi strumenti per la condivisione di modelli. Master's thesis, University of Trento (2002–2003)
18. Visual Paradigm for UML (2007), http://www.visual-paradigm.com/product/vpuml/
19. Common Criteria (2007), http://www.commoncriteriaportal.org/
20. Bozzano, M., Villafiorita, A.: The FSAP/NuSMV-SA Safety Analysis Platform. Int. J. Softw. Tools Technol. Transf. 9(1), 5–24 (2007)
21. Lambrinoudakis, C., Kokolakis, S., Karyda, M., Tsoumas, V., Gritzalis, D., Katsikas, S.: Electronic Voting Systems: Security Implications of the Administrative Workflow. In: Mařík, V., Štěpánková, O., Retschitzegger, W. (eds.) DEXA 2003. LNCS, vol. 2736, p. 467. Springer, Heidelberg (2003)
22. Xenakis, A., Macintosh, A.: Using Business Process Re-engineering (BPR) for the Effective Administration of Electronic Voting. The Electronic Journal of e-Government 3(2) (2005)
23. Howard, M., Pincus, J., Wing, J.: Measuring Relative Attack Surfaces, citeseer.ist.psu.edu/howard03measuring.html
24. Prosser, A., Krimmer, R.: Electronic Voting in Europe - Technology, Law, Politics and Society, Workshop of the ESF TED Programme together with GI and OCG. SchloßHofen / Bregenz, Lake of Constance, Austria, Proceedings, July, 7th–9th, 2004. LNI. GI, vol. 47 (2004)

Compliance of RIES to the Proposed e-Voting Protection Profile

Hugo Jonker[1,2] and Melanie Volkamer[3]

[1] Eindhoven University of Technology
[2] University of Luxembourg
[3] Institute of IT-Security and Security Law (University of Passau)
hugo.jonker@uni.lu, melanie.volkamer@uni-passau.de

Abstract. The RIES-KOA e-voting system was used in the Netherlands as an additional system for the elections by expatriates for the *Tweede Kamer* (roughly: the Dutch House of Commons) elections in 2006. Although the system has been used in other elections in the Netherlands as well, there have been few independent evaluations of the system. In this paper, we apply the recently proposed Protection Profile for e-voting systems to the RIES-KOA system. This serves a two-fold purpose: it is an independent analysis of RIES-KOA and it is the first application of the Protection Profile. We indicate several issues with RIES-KOA and the Protection Profile, respectively, as learned during the analysis.

Keywords: e-voting, RIES, Protection Profile.

1 Introduction

Electronic voting is asserting itself as an addition to the election process. One of the prime examples of this is RIES – the Rijnland Internet Election System. RIES has been developed for regional water management board elections in the Netherlands in order to stimulate voter participation and reduce election costs. RIES was specifically developed to facilitate integration with the existing vote-by-mail system, to allow the voter to choose how she wishes to cast her vote.

After successful use for one election, use of RIES has spread. First to other water management board elections (which are regional organs with regional elections), which piqued the interest of the Dutch government. The government has initiated a project to use RIES to enable expatriates to cast their votes for national elections via the Internet. Previously, expatriates could only cast votes via mail. Past experiences indicated several problems with this medium, mainly that mail is unreliable and slow. Hence, voting officials were interested in providing an alternative which would address these issues, while not deviating too much from the established practice. As RIES was developed to be integrated with a vote-by-mail system, it was the perfect candidate. However, several adaptions to RIES were needed to comply with election law. This had not been an issue before, as water management board elections are not governed by the Dutch election law (as opposed to national elections). Additional changes were prompted by the

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 50–61, 2007.

differences between water management board elections and national elections, for example: In water management board elections, voters choose one person, while in national elections the candidates are affiliated with parties. The effort to adapt RIES, was dubbed "RIES-KOA", where *KOA* stands for *Kiezen Op Afstand* (choosing at distance). The resulting RIES-KOA system was used in the November 2006 national elections for expatriate voting, integrated with the mail-voting system used previously for this purpose. 19,815 votes were cast over the Internet.

This paper focuses on the RIES-KOA system. Its main features are, that it

- requires no special equipment on the voter's side, just secrets sent to the voter via ordinary mail (unlike the Estonian electronic elections, where advanced national ID-cards are available with digital signature capability),
- is simple to use via a web page with all cryptographic functionality hidden in embedded Java script,
- requires no retention of (personal) data by voters to cast their votes,
- provides some kind of verifiability of the outcome, both at an individual and universal level.

RIES is based on academic work and several researchers (see [7]) and international observers (from the OSCE [11]) actively monitor RIES. Despite all that, to date there has been no scientific security evaluation of RIES, and specifically not of the RIES-KOA system. The Dutch e-voting interest group *We Do Not Trust Voting Computers* investigated RIES on a more practical level [9]. This paper provides elements of a scientific evaluation, concentrating on security aspects, together with some recommendations for improvement. The security analysis is based on the Common Criteria Protection Profile (PP) for basic requirements to remote electronic voting systems [14]. Although the PP has been developed in Germany, it is not specific to Germany and hence it is usable for evaluations according to the Common Criteria methodology anywhere.

The remainder of this document is structured as follows: First, the approach used, is described in Section 2. Then, the paper provides a detailed description of the RIES-KOA system in Section 3. The security requirements and assumptions of the PP and are described in Section 4. A synopsis of the security analysis of RIES-KOA is described in Section 5, where RIES-KOA is tested on each security requirement. The paper closes with a discussion of the evaluation results, identifying serious security problems of RIES-KOA as well as suggestions for improvements of the PP.

2 Analysis Approach

In the past, election officials have invited security experts to analyse e-voting systems for vulnerabilities. Usually, each group of experts used their own set of requirements and evaluation methods. Additionally, the depth of evaluation varied much.

This lack of a generic approach has led to conclusions about the security of e-voting systems which are difficult to understand by third parties. To address

this problem, standardized requirements, testing mechanisms, evaluation procedures, and observation techniques are essential. In [14], the authors present a Protection Profile (PP) describing a set of minimum requirements for remote electronic voting, in line with the Common Criteria (CC, [2]) methodology. The PP was written within the context of the BSI (the German Federal Office for Information Security) in cooperation with the e-voting expert group of the German scientific association of Informatics. The PP has been evaluated successfully by the testing authority Security Research & Consulting in the first quarter of 2007, and certification by the BSI is pending. For now, the PP serves as a contribution to the international community and can be used for certification of remote electronic voting systems in any country around the world that has adopted the Common Criteria. Further information about the background is available in [5].

The application of the CC has three advantages compared to existing approaches: First of all, the methodology is known and accepted world-wide. Secondly, the evaluation depth is clearly defined. The third advantage is the possibility to compare different systems based on their evaluation report. Moreover, the PP does not just provide another list of requirements but it combines requirements from other catalogues such as [3,6,1,4,13,10].

The PP for remote electronic voting defines only a basic set of security requirements, which are supported by several assumptions. Here we do not discuss the validity of these assumptions in general, nor their validity in the Dutch situation. The analysis evaluates the RIES-KOA system according to the requirements of the PP. There are some remarks on the general appropriateness of the assumptions in Section 5.4.

3 The RIES System

RIES-KOA operates in three phases: the pre-election phase, the election phase and the post-election phase. Because of the structure and the alignment of the Protection Profile, only the functionality used in the election and post-election phases are analysed. Nevertheless, below we explain all three phases.

The main parties in the RIES-KOA system are: V_i (the i^{th} voter), the voting server, the tallier, and the election authority. Voters choose 1 candidate from the N candidates. Candidates are ranged over by $j, 1 \leq j \leq N$.

In the **pre-election phase** the election authority generates for each voter V_i voter credentials (anonymous, secret values):

- a unique identifier: $sk_i(0)$.
- unique values $sk_i(j)$ for each of the N candidates, where $sk_i(j)$ is derived in a deterministic way from the election authority's master key K and the identifier $sk_i(0)$. These values represent the voter's possible choices.

Note that the credentials are not linked to any specific voter, but only to $sk_i(0)$. The credentials are then printed on official election paper and sealed. This sheet is put into an envelope together with an instruction booklet (which among others directs voters to the RIES-KOA web site to cast their votes). Finally,

these sealed envelopes are addressed using the election register (hence linking credentials to voters) and sent to the voters. So, if the process is implemented correctly no one knows which row was sent to which voter.

Before the vote casting phases starts, the election authority commits to the values $sk_i(j)$ by publishing a list on the voting server, consisting of $N+1$ elements $hash(sk_i(j))$, for $0 \leq j \leq N$, for each V_i. The table with the secret values is stored securely by an official notary. The list thus carries per voter an ordered list of hashes of all possible votes (= encrypted secret values) the voter can choose. Voters can verify that the received values constitute genuine, authentic voting material, by hashing the received $sk_i(j)$ to derive $hash(sk_i(j))$, $0 \leq j \leq N$ and comparing these to the listed values.

In the **election phase** the voter visits the election web site and the browser shows her the ballot. She makes her choice and enters her secret identifier, $sk_i(0)$, and her choice, $sk_i(j)$ (the voter-specific secret value denoting candidate j). The pair $(sk_i(0), sk_i(j))$ is transmitted to the voting server over an SSL channel.

A vote will only be accepted if the pair $(sk_i(0), sk_i(j))$ indeed corresponds to hashes previously published on voting server and no vote for $sk_i(0)$ has been cast before (note that $hash(sk_i(0))$ is the public identity of the voter). After casting the vote, the voter receives a confirmation message.

In the **post-election phase**, the tallier computes and publishes the election results. The votes cast, represented by the pairs $(sk_i(0), sk_i(j))$, are published now on the voting server. Each voter can check that her vote has been counted, that is, appears on the voting server in the list of votes as well as whether the tally is correct. This is achieved by using the pre-election committed list and the $hash()$ function to link cast votes $sk_i(j)$ to candidates j.

4 Security Requirements

The security requirements are described in the Protection Profile [14]. The security objectives defined in the Protection Profile are restated below. They are labeled for cross-referencing in the analysis.

- *AuthorisedVoter.* Only voters eligible to vote who are unmistakeably identified and authenticated by RIES-KOA may cast a vote.
- *NoProof.* No data that RIES-KOA makes available to the voter can be used by the voter to prove her vote to any third party.
- *IntegrityMessage.* RIES-KOA must verify that the content of the authentication message, identification data, ballot data, vote records and the confirmation cannot covertly be deleted, inserted, replayed or amended during transmission (between the client-side and server-side RIES-KOA).
- *ElectionSecrecy.* RIES-KOA must guarantee the election secrecy during transmission; in other words it is not possible to link the voter to her cleartext ballot. In particular, no conclusions about whether the vote is valid or invalid can be drawn from the number or size of the exchanged messages.
- *SecretMessage.* RIES-KOA must guarantee the secrecy of identification data, of the contents of the authentication message and of the vote during

transmission. This is necessary to ensure that an intruder observing the network cannot calculate intermediate results.

- *after-Integrity.* RIES-KOA must guarantee that the polling period data and the result are stored securely within RIES-KOA once the vote count has taken place. Any changes to these data must be recognisable as such.
- *after-ElectionSecrecy.* The system must prevent the possibility to determine how any specific voter voted after votes have been count using the interfaces provided by RIES-KOA – even when supplementary data such as decryption keys are available. A link between voter and vote cannot be inferred from the order and/or time of storage of votes in the ballot box.
- *CancelVote.* The client-side of RIES-KOA must offer the voter the possibility to interrupt the voting process and to retain her right to vote when doing so.
- *EndElection.* RIES-KOA must guarantee that the election committee does not accidentally stop the elections before the official election end time. Following an explicit confirmation, the election committee is able to end the elections prematurely.
- *after-BallotBox.* RIES-KOA must guarantee that its interfaces do not accept votes after the election is closed.
- *AnonElectionCommittee.* RIES-KOA must guarantee election secrecy for all interfaces it provides on the election server during the polling period including the vote count. The election committee is not able to link voters to their plain-text votes using any interface provided by the RIES-KOA system.
- *IntegrityElectionCommittee.* RIES-KOA must not provide any interface to the election committee to insert votes into, delete votes from, or amend votes in the ballot box. In particular, there is no interface of RIES-KOA that allows the election authority to reset RIES-KOA to its original state once an election has started. The RIES-KOA interfaces must guarantee that the election committee cannot allow any voter to cast more than one vote and that the election committee cannot change the authentication data in the list of eligible voters nor change the ballot data. RIES-KOA must guarantee that a restart is not possible once the election has been closed.
- *SecretElectionCommittee.* The election committee interface of RIES-KOA must not provide any knowledge of the content of authentication messages. RIES-KOA must not provide any interface to calculate intermediate results on the voting server.
- *OverhasteProtection.* RIES-KOA is only allowed to accept a vote if the voter has explicitly double-checked and confirmed her vote. To this end, the vote is shown to the voter once again for final verification before the casting is definitive.
- *Correction.* RIES-KOA must place no limit on the number of corrections a voter can make to her vote before she definitely casts it. The voter can correct the vote after the vote has been displayed for final verification.
- *Confirmation.* RIES-KOA must allow the voter to check whether her vote has been stored in the ballot box. This means that the voter is presented with an on-screen confirmation once the vote was successfully stored in the

ballot box. Further, if a voter logs in again, the successful storage of her vote is confirmed on-screen.
- *Malfunction.* The election committee must be able to recognise any malfunction on the server-side RIES-KOA by application of a self-test. After a break-down or other problems, the election committee must have the possibility to start a secure rerun of the system.
- *Log.* RIES-KOA logs the events
  - Storage of the election data at the start of the election,
  - System errors as well as other reductions in the operability of the server-side RIES-KOA,
  - Interruptions of communication,
  - Start and rerun of the election on the server-side RIES-KOA,
  - Closing of the election,
  - Start of the vote count,
  - Determination of the vote count result;

  and allows the election committee to view them.
- *OneVoterOneVote.* RIES-KOA must guarantee that nobody can cast more than one vote and that nobody loses their right to vote without having cast a vote. RIES-KOA must guarantee the right to vote especially in the case of an abort. This can be caused by a voter on the client-side, the client-side itself or the IT environment of the client-side. RIES-KOA must also guarantee that where malfunctions to the server-side occur, as well as to any subsequent restart and the execution of such restart, no data are lost and nobody loses their voting right or is allowed to cast more than one vote. Election secrecy must be preserved in all these cases.
- *AuthElectionCommittee.* RIES-KOA must possess an authentication function that supports separation of duty between a minimum of two members of the election committee. Starting or ending the online election as well as initiating a restart must require two or more members of the election committee to be logged on. Initiating the vote count must also be conditional upon the same requirement being fulfilled.
- *StartVoteCount.* RIES-KOA must guarantee that the election committee is only able to initiate the vote count once the elections are closed.
- *VoteCount.* RIES-KOA must guarantee that all vote records, that are stored in the ballot box after the elections are closed, are correctly evaluated (and, where necessary, correctly decrypted) and contribute to the result of the vote count.

Additional security requirements that the PP does not require of RIES-KOA but of RIES-KOA's environment are covered by the following assumptions:

- Election data is properly and correctly installed on RIES-KOA before the start of a polling period; the ballot box is empty; the election preparation phase has been carried out correctly; and RIES-KOA is correctly initialised.
- The voter ensures that nobody is watching her while she votes.
- The election committee accesses no data other than that on RIES-KOA; i.e., it uses only the functions made available by the RIES-KOA system.

- The voter handles her voting credentials with care and is consistent in doing so; in particular, she ensures these remain private (solely accessible by herself).
- The voter acts responsibly in securing the client device. This includes the assumption that the voter does not manipulate her client device.
- The election server is protected against network attacks.
- The election server and the network are assumed to be robust, to be available and to provide a sufficient level of quality of service.
- No one outside the election committee, as appointed by the election organiser, has access to the server room, nor to the election server for the duration of the polling period, until the vote count.
- The data storage hardware is functioning correctly.
- The correct time is made available by the server's IT environment.

As can be inferred from the descriptions above, the PP has a limited scope. The PP only takes the voting period into account (including the counting process) and is focused on a set of basic requirements for usability and security. As such, compliance to the profile does not imply that the evaluated voting system is a secure system – it means that the evaluated voting system satisfies the set of basic requirements in an environment where the given assumptions hold.

5 Security Analysis

The analysis evaluates the compliance of RIES-KOA to the proposed Protection Profile [14]. In a full-blown Common Criteria evaluation, adherence to the security functional requirements would be checked. However, as these requirements are derived from the security objectives, the below analysis employs the security objectives, which are more readable.

Note that the scope of the PP is limited. The PP is written for elections adhering to the most basic election principals (universal, free, secret and democratic elections). This type of elections is common for e.g. national elections, however, specific elections may deviate from this norm (e.g. postal voting or voting by share holders). This PP cannot accommodate elections where some of these principals have been relaxed.

5.1 Used Sources

The analysis below is based on the official RIES-KOA documentation [12], augmented by additional insights from personal experience[1]. The official documentation focuses mainly upon describing the operational aspects of the system, such as used file formats. Unfortunately, there is a lack of other publicly available sources of information on the RIES-KOA system, apart from the description of the original RIES system in [7].

[1] Most notably a meeting with ms. Beneder from the RIES-KOA project on 22 February 2007, and a workshop on the security of RIES organised by SURFnet on 15 May 2007.

The analysis is of the conceptual RIES-KOA system, because the number of sources are limited, and the analysis was undertaken *a posteriori*. Nevertheless, as the official RIES-KOA documentation is used, we believe that the below analysis carries over well to the system used in the November 2006 elections.

5.2 Compliance to Security Objectives

Below, the security objectives are listed once more by name, and for each security objective, the compliance of RIES-KOA is analysed. *FAIL* indicates lack of compliance, while *PASS* indicates RIES-KOA meets a particular security objective. *INCONCL* means that the used sources did not provide enough information to determine a PASS/FAIL verdict.

objective	**result**	**objective**	**result**
AuthorisedVoter	PASS[1]	IntegrityElectionCommittee	INCONCL[8]
NoProof	FAIL[2]	SecretElectionCommittee	INCONCL[9]
IntegrityMessage	FAIL[3]	OverhasteProtection	PASS[5]
ElectionSecrecy	FAIL[3]	Correction	PASS[5]
SecretMessage	FAIL[3]	Confirmation	PASS[5]
after-Integrity	PASS[4]	Malfunction	INCONCL[10]
after-ElectionSecrecy	FAIL[2]	Log	INCONCL[11]
CancelVote	PASS[5]	OneVoterOneVote	PASS[12]
EndElection	INCONCL[6]	AuthElectionCommittee	FAIL[13]
after-BallotBox	PASS[5]	StartVoteCount	INCONCL[14]
AnonElectionCommittee	PASS[7]	VoteCount	PASS[5]

1. The RIES-KOA system prescribes that the voter credentials are delivered via post. This is an insecure, unauthenticated channel, and thus the voting credentials must be considered exposed. However, the correct distribution (and handling) of voter credentials is covered by the assumptions. Hence the PASS verdict – note that these assumptions do not hold for the elections in which RIES-KOA was used.
2. The RIES-KOA system is specifically designed to have these proofs.
3. The RIES-KOA system relies on SSL to secure the connection between RIES-KOA and the voter. However, the RIES-KOA documentation does not mention how to configure SSL. An incorrect setup can lead to exposure (as SSL specifically allows the option to use no encryption). Without specifying the SSL setup, we cannot assume that integrity or secrecy will hold.
   Note that with a correct SSL setup, the verdict would be PASS.
4. The set of votes is signed and made publicly available.
5. The RIES-KOA system supports this.
6. The RIES-KOA system documentation mentions that a current election can be closed, but does not describe what preconditions must be met for this status change to occur.
7. Note that this information *is* available on the server and thus to anyone with administrator access to the server. However, this is covered by the assumptions, hence a PASS verdict is in order.

8. RIES-KOA satisfies most requirements listed here, as it has no specific election committee interface. However, RIES-KOA has a provision for activating substitute voter credentials. This procedure violates the spirit if not the precise wording of this requirement.
9. The documentation does not mention when or how the voting server provides the set of collected votes. Specifically, there is no mention of any provision to prevent intermediate result from being calculated from a partial set of votes.
10. The documentation available provides no information on self-tests.
11. The documentation mentions several logs, amongst which a `sysstatlog` and an `eventlog`, but fails to specify exactly what is logged.
12. Note that the PASS relies on the assumption that no voter has access to two (or more) different sets of voter credentials (which follows from the assumptions in the PP).
13. Starting or halting an election requires access to a specific terminal per server used in the election. The available documentation in no way distinguishes users for this purpose.
14. The documentation fails to mention when access to the vote count becomes available.

5.3 Evaluation Conclusions

We reiterate the remark made in the beginning of this Section: note that the PP has a limited scope; care should be taken not to mistake a general PASS verdict for a "secure system".

The most obvious conclusion is that the available documentation of RIES-KOA has some profound gaps. With better documentation, or access to more documentation, the analysis would probably be more positive in some points (more specifically, a description of SSL setup would mitigate *IntegrityMessage*, *ElectionSecrecy* and *SecretMessage*). Note that in the course of a full CC evaluation, one of the largest categories of errors found are documentation errors. In the course of a CC evaluation process, such errors are corrected. Hence, similar omissions in the RIES-KOA documentation do not seem too grave.

A second conclusion is that the design decision to grant receipts to voters to ensure verifiability implies that RIES-KOA fails the *NoProof* objective by design. This is a more serious issue and unnecessary to ensure verifiability (see e.g. [8]). However, this failure stems from a design issue in RIES, the consequences of which are well known to the involved parties.

The foremost conclusion of the analysis, however, is more grave. The documentation has several severe lacunes in addition to the ones mentioned before. There is insufficient information on self-tests, on access to the ballot box, on logging and on preconditions for starting or halting elections. In none of these cases, however, is there any indication in the available documentation that the RIES-KOA system complies with the requirements of the PP.

In more detail, the issues that were discovered were the following:

- The documentation fails to have any mention of self-tests. As self-tests facilitate detection of malfunctions in the system (*Malfunction*), it is unclear (even unlikely) that RIES-KOA will catch malfunctions.
- The documentation does not specify when the ballot box becomes accessible for counting. Hence, intermediate results (*SecretElectionCommittee*) and premature start of vote count (*StartVoteCount*) cannot be ruled out.
- The lack of documentation on logging means that it is impossible to determine whether sytem errors, communication errors, and even status changes such as starting or halting elections can later be traced (*Log*).
- And finally, there is no mention in the documentation of support for multiple users in the system. This means that RIES-KOA lacks the support to ensure elections can only be halted by more than one person (*AuthElectionCommittee*).

We believe these to be serious issues, the extent of which the involved parties may not completely realise. Neither do these seem to be merely documentation omissions. The provided documentation strongly indicates (without being conclusive) that RIES-KOA lacks the functionality needed to support these requirements.

In any Common Criteria analysis, the documentation is bound to be lacking. For those cases, where it is clearly an omission (e.g. SSL setup), this can be easily addressed. However, the documentation on RIES-KOA misses several issues that seem not to have been implemented in the system at all. Even if the questionable design decision of having voter proofs is assumed to be correct, the system (in its current state) still fails to meet the basic requirements as set forth in the PP.

5.4 Evaluation of the Methodology

This analysis brings to light not only several weaknesses of RIES-KOA but also some issues of the PP. Here we note the most important issues.

A lacune in the PP discovered during the analysis is the complete lack of requirements on verifiability. Verification of the result and of inclusion of a vote as cast by a specific voter is one of the foremost issues for generic acceptance with electronic voting – can the voters be convinced that the result is really based on the votes as cast?

A second issue in the PP is the reliance on very strong assumptions. It will be extremely difficult to ensure that none of the assumptions can be violated. The following two assumptions are the main problems in this regard:

- *The election committee [...] uses only the functions made available by the [RIES-KOA system].*

The consequence of this assumption is that the PP only defines security requirements on the interfaces offered by RIES-KOA. However, this can be

insufficient. In the case of RIES-KOA, a database is used to store all data (including votes). Databases often have their own interface. By not covering this interface as well, a voting system compliant with the PP may still allow arbitrary manipulation of votes via a direct database interface. Other software used by voting systems (e.g. operating systems, web servers) can have similar features. Hence the PP should be updated to address the use of third-party software.

– *[. . .] the election preparation phase has been correctly carried out [. . .].*

This assumption covers the generation and distribution of the authentication data to the voter. The intention was that none but eligible voters receive voting credentials. To this end, the voting credentials must remain secret (e.g. they are not readable through the envelope using a bright light source). In systems such as RIES-KOA, however, this phase is essential to election secrecy. The election committee must not know the content of the election material – otherwise they can break election secrecy and (e.g.) cast votes in a voter's stead.

6 Conclusions

We analysed the Dutch RIES-KOA system used in November 2006 national elections for expatriate voting. This analysis was executed according to the Common Criteria Protection Profile (PP) describing basic requirements for remote electronic voting. The PP specifies quite a number of assumptions on the environment. The evaluation was done under the premise that these assumptions hold. Surprisingly, even given these relaxation of constraints on our part (the evaluation assumed that the system's environment indeed satisfies the assumptions of the PP), RIES-KOA cannot successfully meet the requirements of the PP. RIES-KOA fails to meet several security objectives outright, and might fail several more where we had to conclude "inconclusive".

The analysis also brought to light several limitations of the PP. On the whole, we can conclude that the PP enables a structured evaluation of remote electronic voting systems. The current PP should be viewed as an initial version outlining the most basic requirements. The PP does not (yet) capture all security aspects of e-voting as desired for most types of elections (such as verifiability). To achieve this, the PP needs to mature further.

Nevertheless, even this early version of the PP has already found issues in RIES-KOA. This underlines the need for a structured approach to security in e-voting, as offered by the PP.

Acknowledgments

We are very grateful for the insightful discussions on RIES with Berry Schoenmakers and Bart Jacobs. Funding for Melanie Volkamer's research visit at TU/e was provided by the German Academic Exchange Service (DAAD).

References

1. Voting standards: Project 1583 - voting equipment standard, project 1622 - electronic data interchange (2005)
2. Common Criteria for Information Technology Security Evaluation, Version 3.1 (2006)
3. Council of Europe. Legal, operational and technical standards for e-voting. Recommendation Rec(2004)11 adopted by the Committee of Ministers of the Council of Europe and explanatory memorandum. Council of Europe, Straßburg (2004)
4. Gesellschaft für Informatik, e.V.: Anforderungen an internetbasierte vereinswahlen. Informatik Spektrum 28(5), 432–435
5. Grimm, R., Krimmer, R., Meißner, N., Reinhard, K., Volkamer, M., Weinand, M.: Security requirements for non-political internet voting. In: Proceedings of the 2nd International Workshop on Electronic Voting. LNI 86, pp. 203–212 (2006)
6. Hertmann, V., Meißner, N., Richter, D.: Online Voting Systems for Nonparliamentary Elections - Catalogue of Requirements. Technical report, Physikalisch-Technische Bundesanstalt Braunscheig/Berlin (8.5.2004)
7. Hubbers, E., Jacobs, B., Pieters, W.: RIES - internet voting in action. In: COMPSAC (1), pp. 417–424 (2005)
8. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Atluri, V., De Capitani di Vimercati, S., Dingledine, R. (eds.) WPES 2005: Proceedings of the 2005 ACM workshop on Privacy in the electronic society, pp. 61–70. ACM Press, New York (2005)
9. Kruijswijk, L.: i-voting with RIES analyzed (November 17, 2006)
10. German Ministry of the Interior (BMI). Verordnung über den Einsatz von Wahlgeräten bei Wahlen zum Deutschen Bundestag und der Abgeordneten des Europäischen Parlaments aus der Bundesrepublik Deutschland. Bundeswahlgeräteverordnung (BWahlGV) (Vom 03.09.1975 (BGBl S. 2459) zuletzt geändert am 20.04.1999 (BGBl I S. 749)) (1999)
11. OSCE. The Netherlands parliamentary elections (22 November 2006) (March 12, 2007)
12. Pont, P.M., Hannink, A., Hoeienbos, J., Rijkschroeff, M., Schuurman, J.: RIES-KOA – functioneel ontwerp (November 13, 2006)
13. Volkamer, M., McGaley, M.: Requirements and Evaluation Procedures for eVoting. In: Dependability and Security in e-Government (2007)
14. Volkamer, M., Vogt, R.: Protection Profile - Central Requirements for Online Voting Systems. Technical report, German Research Center for Artificial Intelligence (2007)

Compliance of POLYAS with the BSI Protection Profile – Basic Requirements for Remote Electronic Voting Systems

Kai Reinhard and Wolfgang Jung

Micromata ltd., Marie-Calm-Str. 3, 34131 Kassel, Germany
polyas@micromata.de
http://www.polyas.com

Abstract. In the past year and a half a group of experts in electronic voting developed a Common Criteria Protection Profile describing basic requirements for remote electronic voting systems. This work was lead-managed by the German Federal Office for Information Security (BSI) and the German Research Center for Artificial Intelligence (DFKI) and initiated by the German Gesellschaft für Informatik (GI - society for informatics). To complete this work Micromata's POLYAS system, which is used for the GI elections needs to be evaluated against this Protection Profile. As a first step a high-level evaluation based on the security objectives has been carried out. The results are presented in this paper.

1 Introduction

With the emergence of the Internet as a worldwide available data network, the idea of using it for elections was born. But while e-commerce and e-government took off right away it took a long time for real elections to emerge. The public first recognized remote electronic voting was the Democratic Primaries election in Arizona in 2000.

In this context, the remote electronic voting system POLYAS [POLY07] has a long-standing history: As early as 1996 the first election was carried out with 64,000 young Finnish people in a project supported by the Finnish Ministry of the Interior. To date, more than 340,000 votes were cast in total using the system; 210,000 of which in Germany. POLYAS is one of the most experienced remote electronic voting systems in Europe. Part of its success is the close partnership with the German Gesellschaft für Informatik (GI- society for informatics) where POLYAS was used to complement and partly replace the postal voting process for their yearly held elections [KRVo05]. Further, the GI founded an advisory board of security experts from all over Germany to evaluate and improve POLYAS over nearly three years.

One outcome of this work was the realization that remote electronic voting software needs proper validation and formal evaluation from a third party. For this the board proposed the use of Common Criteria methodology and the development of a Protection Profile, which was introduced in 2005 and is currently

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 62–75, 2007.

under certification [GKMR06]. This paper is to be seen in the light of these developments.

We will present the functionalities of POLYAS, test its compliance with the Protection Profile (PP), and give an assessment.

2 Analysis Approach – The Protection Profile

In Germany during the last year and a half, a Protection Profile (PP), "the basic requirements for remote electronic voting systems" [PP07] following the methodology of the Common Criteria (CC) [CC06] has been developed by a group of experts in electronic voting lead-managed by the German Federal Office for Information Security (BSI) and the German Research Center for Artificial Intelligence (DFKI). Currently, the PP has been successfully evaluated but the certification by the Federal Office for Information Security (BSI) is still missing, however, its completion is expected towards the end of September. After the successful evaluation of the PP, the next step is to evaluate remote electronic voting systems against it. We aim to be the first having a voting system being evaluated using the Common Criteria and in particular against this Protection Profile. Before we formally start the evaluation we ask an institute to check on a high level whether POLYAS passes the evaluation or whether changes are necessary. The results are presented in the analysis section.

To understand this analysis, we shortly explain how the Common Criteria and the Protection Profile works: Amongst other things the PP contains a list of security objectives, which are mandatory as basis requirements for remote electronic voting systems. During the study, an analysis for all security objectives has been carried out to establish whether (and if yes, how) POLYAS meets these objectives. The PP includes a few assumptions in relation to the environment in which the voting system needs to be used. Thus, the election is only secure if a system passes the respective security objective and is used in the context of the environment outlined. Such assumptions in [PP07] are e.g.:

- any election data is properly and correctly installed before the starting the election,
- the election committee accesses no data other than that using the interfaces provided by the system;
- no one else besides the election committee has access to the data.

We use the assumptions of the PP as quoted. A discussion of the purpose and aim of these assumptions in direct relation to the PP is beyond the scope of this paper.

3 POLYAS System Description

3.1 Overview

According to the classification in [VoKr06], POLYAS belongs to the following classes: First of all it is a *Remote Electronic Voting System*, also called Internet

Voting System. The identification and authentication technique in use is *secret-based*, i.e., the voter already knows his identification code (ID) in advance because it is, e.g., his membership or student number and the authentication token, also called "voting TAN" is sent via ordinary mail. With respect to the technique in use to ensure electoral secrecy. POLYAS belongs to the class of systems where *anonymity is ensured in the voting phase.* The particular sub-class POLYAS belongs to is the one based on a *separation of duty principle*, i.e., there are at least two different voting servers whilst one server checks the voter's eligibility to cast a vote and sends a random voting token to the voter and the second server can then later identify eligible vote messages containing a valid voting token. The voting clients used within POLYAS belongs to the *thin-client* class, i.e., the voter does not need any particular hardware or software to cast a vote but only an arbitrary web browser including the text-based web browser lynx.

More detailed, POLYAS is composed of three voting servers: the electoral register server (ERS), the validator server (VS) and the ballot box server (BBS). The voter using his voting client (VC) only communicates with ERS and BBS. During the voting phase the only one communicating with VS is the BBS. The VS mainly controls the other two servers. All three servers are located at three different places.

3.2 Process in Three Phases

Pre-Voting Phase – The following six main steps can be identified:

(1) Generation of the voting TAN: A list of voting TANs are generated in such a way that the TANs are only available in a hashed ($hash(TAN)$) and encrypted ($encr(sk_P, TAN)$) manner. The hashed values are stored in an electronic copy of the electoral register which is later installed on the ERS. The encrypted values are also stored in an electronic copy of the electoral register, which is then sent to a service provider, which prints the election material. Whilst the first electoral register contains the ID of the voter, the second contains the corresponding name and postal address.

(2) The service provider allocates a decrypted voting TAN to each piece of election material. The TAN is covered and needs to be revealed by the voter in order to read it. According to this procedure the TANs are encrypted with the public key of this service provider (note: the service provider does not know the voter's ID number).

(3) For each of the three servers, a https, a communication and a database key pair are generated. The https public keys are made public. The private communication and database keys are then encrypted and one pass phrase for each of the keys is needed. Thus, there are six pass phrases, each one if them being handed over to six different members of the election commission. Consequently, we have three pairs of members within the election commission and each one of these groups is needed to start off the election on one of the servers.

(4) The private communication keys of ERS sk_{ERS} and VS sk_{VS} are used to sign the hashed TANs from the electoral register. Thus each column contains:

$$ID - hash(TAN) - sig_{ERS} - sig_{VS} \tag{1}$$

where

$$sig_{ERS} := sig(sk_{ERS}, hash(TAN)) \text{ and } sig_{VS} := sig(sk_{VS}, sig_{ERS}) \quad (2)$$

The entire electoral register is then installed on the ERS. Moreover, the whole electoral register is hashed and signed with the private communication key sk_{ERS}. This signed register is stored externally in a secure place.

(5) For each of the three servers two remote access tokens are generated both of which need to be entered in order to get access to the corresponding server (each pair of commission members is responsible for a particular server). Once again, six secrets are to be distributed amongst the election commission.

(6) The three servers are configured and the corresponding POLYAS Software for each of them is installed (including the ballot information on the BBS).

Voting Phase – First of all the POLYAS software on the BBS as well as on the VS is started off. Finally, at the official beginning of the election the software on the ERS is started off. This is done by going to a particular web page and entering the pass phrases to decrypt the database and the communication key of each of the three servers.

The high-level protocol steps during which a voter casts his vote are described in figure 1.

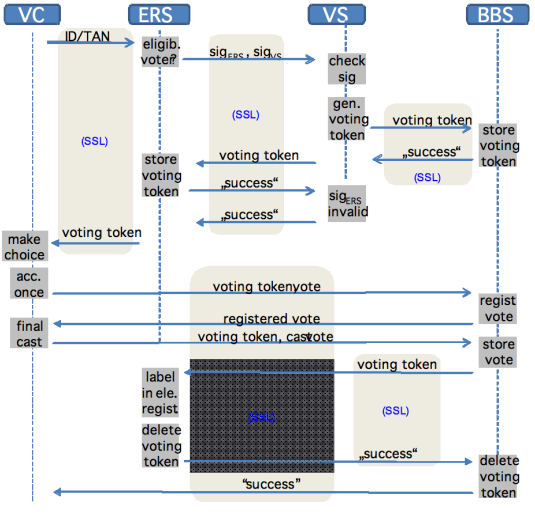

Fig. 1. POLYAS Protocol

The specialty of this protocol is that the vote is already sent to the BBS after the first acknowledgement. Here, the vote is already registered and then the registered vote is sent back to the voter again. The motivation for choosing

this process is that the voter can be sure that the BBS has interpreted his vote properly. In general, votes are stored in an encrypted and signed manner. Moreover, the voting tokens are also stored and encrypted using the public key of the involved database. The votes are stored in a randomized order in blocks of 30 on the BBS (whilst 30 is variable). As soon as a block is completed, these blocks are hashed and published as a hash-chain in the following way:

$$hash(permut(vote_1, ..., vote_{30})) \tag{3}$$

$$hash(hash(permut(vote_{31}, ..., vote_{60}))\#hash(permut(vote_1, ..., vote_{30}))... \tag{4}$$

The communication between the servers is secured by signed and encrypted messages (using the corresponding communication keys) using SSL.

Result Calculation – To close the election, first the VS is taken offline, and then the other two servers. To do so, for each server those two members knowing the secrets to access the server need to be present. After having taken the servers offline the member of the election commission knowing the pass phrase for the secret database key of the BBS needs to enter this pass phrase in order to start the counting procedure.

4 Analysis Against the Protection Profile

This section analyzes the POLYAS system against the Protection Profile (PP). The list of security objectives was retrieved from the current version of the PP. In those instances, where security objectives (SecObj) contain more than one issue to be analyzed, the objective has been segmented to ensure that all aspects are covered. As a consequence, the amount of the SecObj differs from those in the PP. For each security objective it is outlined whether (and if yes, how) POLYAS meets the requirements. There is a distinction between **PASS** and **FAIL** and in addition to that **ORG** has been introduced to indicate that the relevant persons are aware of any problems or attacks, but currently only have come up with an organizational solution to meet the requirements of the corresponding security objective.

The main problem for deciding upon pass or fail appeared for those security objectives related to the election commission interface because POLYAS provides currently only an election commission interface to start off the election. However, for all other functionalities a remote access (with two people) to one of the servers needs to be established, and then a functionality such as *stop the election or start the counting* is initiated using shell commands. Moreover, for organizational purposes the access control to access the servers is carried out in the CC speech because it is based on the operating system running on the server, which is not part of the evaluation. For this particular analysis this access control is considered as part of the POLYAS systems knowing that it could correspondingly be implemented here.

> *SecObj1: Only voters eligible to vote who are unequivocally identified and authenticated by the voting system may cast a vote.*

PASS. It is only possible to get a voting token enabling a voter to cast a vote after sending the ID and the voting TAN to the ERS, which decides whether the request comes from an eligible voter who has not yet cast a vote. Without having such a valid voting token you can send vote messages to the BBS but these are rejected.

At first glance, it looks like the pair of commission members responsible for the ERS could start the protocol with step 2 without a voter trying to log in. But this scenario is covered with the assumption that the election commission only uses the POLYAS interface, which of course does not provide such functionality. Similarly, an attack is covered with the same assumption: if a voter logs in but does not complete his voting process, the ERS knows this voter's ID and voting TAN (not the hashed one) and once the election commission members have access to these values, they can cast a vote on behalf of the voter. But again this is not possible using only the POLYAS interface.

SecObj2: No data made available to the voter by the voting system can be used by the voter to prove his vote to any third party.

PASS. According to figure 1 the voter only receives the following types of messages: "voting token," "ballot," "registered vote," "success." In the first run it might look as if this trace is a proof; it is not because the voter might have changed his choice after having received the "registered vote" message. In this case, the original trace was: "voting token," "ballot," "registered vote", "registered vote"', "success." Thus, such a trace cannot be used as proof.

SecObj3: The voting system must verify that the content of the authentication, identification, ballot, vote and confirmation message cannot be deleted, inserted, replayed or amended during transmission without being detected.

PASS. The integrity of all transmitted messages is secured by SSL between the servers by encrypting and signing the messages using the corresponding communication keys.

SecObj4: The voting system must guarantee election secrecy during transmission; in other words it is not possible to link the voter to his clear-text vote, also not to the information whether the vote is valid or invalid concluded from the number or size of the exchanged messages.

FAIL. First of all, the vote is transmitted encrypted via SSL. Secondly, the vote is not sent together with the identification data, not even during one SSL session. Thus, one can only link the encrypted identification data to the encrypted vote via corresponding sender IP addresses. The current problem is that someone who is observing the Internet and knows, which IP-address a particular voter has, can limit the possible choices the voter makes because of the size of the vote message. Especially, casting an invalid vote by choosing all candidates is observable.

SecObj5: The voting system must guarantee the secrecy of identification data, of the contents of the authentication message and of the vote during transmission (to prevent the calculation of intermediate results).

PASS. The secrecy of all transmitted messages is secured by SSL.

SecObj6: The voting system must guarantee that the polling period data and the results are stored securely after the result calculation.

ORG. After the completion of the result computation, POLYAS computes a hash value of the electoral register (including those who cast a vote and who did not) and a hash value of all votes. These two hash values are printed immediately and are part of the election commission documentation, which is signed by the election commission.

SecObj7: It must not be possible to determine how any specific voter voted after the votes have been counted using the interfaces provided by the voting system - even when supplementary data such as decryption keys are available. A link between voter and vote cannot be inferred from the order and/or time of storage of votes in the ballot box.

PASS. The only link between a voter and his vote on the server side is the voting token. But the voting token is deleted at the ERS and the BBS just after completing the voting process for the corresponding voter. Thus, even knowing all data from the servers after the election it is not possible to break the election secrecy because the link was already removed during the election.

SecObj8: The client-side of the voting server must offer the voter the possibility to cancel his voting process and to retain his right to vote when doing so.

PASS. The voter can cancel his voting process at any time by closing the web browser. Moreover, there is a particular button to cancel the voting process.

SecObj9: The voting system must guarantee that the election committee does not accidentally stop the election before the official election end time. The election committee must be able to end the election before the official end.

ORG. At the particular day and time the election commission meets in order to first deactivate the VS and the BBS and later the ERS. But it is not controlled by POLYAS whether the end of the election is already reached.

SecObj10: The voting system must guarantee that it does not accept votes after the election is closed.

PASS. After the deactivation of all three servers, they are not online anymore. If a voter still tries to cast a vote, he/she will get the information that the election is over.

SecObj11: The voting system must guarantee election secrecy for all provided interfaces on the election server during the polling period including the vote count.

PASS. First of all, there is no functionality implemented for the election commission to access the (encrypted) votes, thus it is not possible to break the election secrecy using the POLYAS interface. Together with the assumption that the election commission only uses the functionality provided by the TOE, this security objective is ensured. However, POLYAS provides even more security. Here, the assumption is introduced that the two pairs of commission members (i.e. the one that has access to the BBS and the other one with access to the ERS) do not cooperate. However, it is assumed that each one of them is corrupt and accesses the voter database and the electoral register respectively. The set up is as such that the BBS knows only the votes but does not have any other voter related information like the voter's ID for instance; and the ERS knows which voter is in the voting process but does not get any information about the voter's choice. If, however, one of the pairs accessed the electoral register in order to obtain information on what time a particular voter cast his vote for instance, and the other pair accessed the vote database to get the corresponding time information about incoming votes, both pairs could cooperate to link the voter to his vote.

SecObj12: The voting system must not provide an interface to the election committee to insert votes into, delete votes from, or alter votes in the ballot box.

PASS. First of all, there is no functionality implemented for the election commission to access the database containing the (encrypted) votes (other than for the result computation); thus, it is not possible to add, change or alter cast votes using the POLYAS interface. Together with the assumption that the election commission only uses the functionality provided by the TOE, this security objective is ensured. However, POLYAS provides even more security. Here the assumption is introduced that the pair of election commission members responsible for the BBS together with the one knowing the pass phrase for the BBS's communication key do not access the database in order to replace incoming votes by new ones. Moreover, it is necessary to assume that neither one of them (one having access to the BBS and the other one to the ERS) do cooperate. On the grounds of this second assumption, is not possible to delete or add votes undetected because then the number of votes is not equal to the number of electors in the electoral register. The first assumption is necessary because generally speaking altering votes is possible by accessing the database but because of the hash chains only less than 30 votes (depending on the point in time) can be manipulated because changing the other ones would be detected thanks to the published hash chains. Note: it is not enough to have access to the database but also to the private database key of the BBS to sign the changed vote.

SecObj13: The voting system must not implement an interface functionality to reset the system to its original state once an election has started.

ORG. This security objective is only ensured by organizational measures because all 12 different members of the election commission need to be present in order to reset the system.

SecObj14: The voting system must not implement an interface functionality to allow the election committee to give any voter the chance to cast more than one vote.

PASS. First of all, there is no functionality implemented for the election commission to access the electoral register in order to shift a "has cast a vote" label to a "has not yet cast a vote" label, thus it is not possible to enable voters to cast more than one vote using the POLYAS interface. Together with the assumption that the election commission only uses the functionality provided by the TOE, this security objective is ensured. However, POLYAS provides even more security. It is assumed that the election commission members being responsible for the ERS and its communication key pass phrase do not cooperate with those responsible for the VS in order to allow particular voters to cast more than one vote. This assumption is sufficient because if the electoral register has been changed, the VS notices corresponding voting processes because it stores which voting TAN (exactly which hashes TAN) has already been in use and will cancel those voting processes.

SecObj15: The election committee must not be able to change the authentication data in the list of eligible voters nor change the ballot data using the system's interface.

PASS. First of all, there is no functionality implemented for the election commission to access the electoral register or the ballot information during the election phase, thus it is not possible to alter either the authentication data or the ballot information using the POLYAS interface. Together with the assumption that the election commission only uses the functionality provided by the TOE, this security objective is ensured. However, with respect to the voting token POLYAS provides even more security. The evaluation is based on the assumption that the pair of election commission members responsible for the ERS does not cooperate with the pair responsible for the VS in order to change the voting token. This assumption is sufficient because if the members responsible for the ERS change the electoral register, they willnot be able to generate valid VS signatures on the hash value of these voting tokens. The VS is able to notice a voting process based on a change TAN and would therefore cancel it.

SecObj16: The voting system must guarantee that a restart is not possible once the election is closed.

ORG. This security objective is ensured only through organizational measures because all 12 members of the election commission need to be present in order to reset the system.

SecObj17: The voting system must not provide any knowledge of the content of authentication messages.

PASS. First of all, there is no functionality implemented for the election commission to access the electoral register, thus it is not possible to read the authentication data using the POLYAS interface. Together with the assumption that the election commission only uses the functionality provided by the TOE, this security objective is ensured. However, with respect to the voting token POLYAS provides even more security. Here the assumptions introduced above are not necessary because the voting token is only stored in a hashed manner.

SecObj18: The voting system must not provide any interface to calculate intermediate results on the voting server.

ORG. This security objective is only ensured with appropriate organizational measures because generally speaking it is possible to calculate intermediate results on the voting server, however, three different members of the election commission, in particular those two who dispose of the pass phrases to access the BBS and the one knowing the pass phrase, are required to decrypt the private database key of the BBS.

SecObj19: The voting system must not only accept a vote if the voter has explicitly double-checked and confirmed his vote. To this end, the vote is shown to the voter once again for final verification before the casting is final.

PASS. Following the step "choosing candidates" is the step "display vote." It is reached by pressing the button "check vote cast." Now the voter can either change his vote or proceed to the final casting (of the displayed vote) by pressing the corresponding button.

SecObj20: The voting system must not place any limit on the number of corrections a voter can make to his vote before he/she finally casts it. The voter must be able to correct his/her choice after the vote has been displayed for final verification.

PASS. First of all, the voter is able to change his/her choices in the "choosing candidate" step as often as desired. Moreover, after having pressed the "check vote cast" button, the voter has the opportunity to go back to the step "choosing candidate" and change his choice again.

SecObj21: The voting system must allow the voter to check whether his vote has been stored in the ballot box. This means that the voter is presented with an on-screen confirmation once the vote has successfully been stored in the ballot box.

PASS. If the voter accepts the displayed vote and has pressed the "cast vote" button, the vote is stored in the ballot box database and the corresponding voting tokens are deleted on both servers. If all these steps are completed, the voter gets a receipt notifying him that his vote has been stored successfully.

SecObj22: If a voter logs into the voting system after having cast a vote, the successful storage of his vote must be confirmed onscreen.

PASS. If the voter logs into the ERS after having cast a vote, he receives a message telling him that his vote has already been cast and stored successfully.

SecObj23: The election committee must have the possibility to recognize any malfunction on the server-side of the voting system by using a self-test.

PASS. Before the election each part of the software is digitally signed, meaning at any time the two election commission members responsible for a particular server can access the server and check whether the software running is still the one that has been installed. Moreover the servers are observed using the Nagios software. This software checks regularly whether the server and the databases are still online and available.

SecObj24: After a break-down or other problems, the election committee must have the ability to start a secure rerun of the system.

PASS. A comprehensive and exhaustive recovery concept has been developed containing all possible breakdown and restart scenarios. In case of system breakdowns including data loss the election commission is informed and possible actions are discussed (is a restart possible?).

SecObj25: The voting system must log the events: (1) storage of the election data at the start of the election, (2) system errors as well as other reductions in the operability of the server-side software, (3) interruptions of communication, (4) start and rerun of the election on the server-side software, (5) closing of the election, (6) start of the vote count, and (7) determination of the vote count results. Moreover, the election committee interface must provide the possibility to view the log files.

FAIL. Most of the events listed above are logged by POLYAS. The election data stored at the beginning of the election and the results after the counting process are missing in the current version. The log files can be read on the corresponding server.

SecObj26: The voting system must guarantee that nobody can cast more than one vote and that nobody loses their right to vote without having cast a vote.

PASS. The POLYAS software installed on the ERS ensures that only those voters having valid IDs and voting TAN can continue the voting process and then cast a vote. It also ensures that all such voters can continue the voting process.

SecObj27: The voting system must guarantee the right to vote especially in the case of an abort. This can be caused by a voter on the client-side, the client-side itself or the IT environment of the client-side. The voting system must also guarantee that if malfunctions at the server-side occur or during the execution of any subsequent restart, no data are lost and nobody loses their voting right or is allowed to cast more than one vote. Election secrecy must be preserved in all these cases.

PASS. During the development phase a possible breakdown in each step of the protocol has been discussed and investigated with respect to losing the right to vote or having the possibility to cast more than one vote. The protocol has been modified in a way that the one voter-one vote principle can be ensured for all these situations as long as the voter takes care that in the event of not having received the final receipt, he or she needs to re-login to complete the voting process.

SecObj28: The voting system must ensure that the election secrecy is ensured for the restart.

PASS. Because there are no data stored on the servers, it is not possible to link a voter to his vote, not even during a recovery procedure. Depending on the time on the protocol when the problem appears, there might be a slight off-chance that there is still information on the link "voter - voting token" on the ERS and the information on the link "voting token- encrypted vote" on the BBS but - to link the information it is necessary that both member pairs cooperate.

SecObj29: The voting system must possess an authentication function for the election commission that supports separation of duty between a minimum of two members of the election committee. Starting or ending the online election as well as initiating a restart must require two or more members of the election committee to be logged on. Initiating the vote count must also be conditional upon the same requirement being fulfilled.

PASS. It is only possible to access any of the servers if both members responsible for the corresponding server are present (one has to enter the pass phrase and the other has to enter the encrypted key).

SecObj30: The voting system must guarantee that the election committee is only able to initiate the vote count once the elections are closed.

ORG. This security objective is ensured only through organizational arrangements because generally speaking it is possible to initiate the vote count before the election is closed, however, POLYAS requires three different persons to do so. These have to be three different members of the election commission, in particular those two who dispose of the pass phrases to access the BBS and the one

who is knowledgable of the pass phrase to decrypt the private database key of the BBS.

> *SecObj31: The voting system must guarantee that all vote records which are stored in the ballot box after the elections are closed are correctly evaluated (and, where necessary, correctly decrypted) and contribute to the result of the vote count.*

PASS. The source code has been examined by the Physikalisch-Teschnische Bundesanstalt (PTB). They especially checked the vote casting algorithm.

5 Conclusion

This paper initiates the preparation for the evaluation of POLYAS against the Protection Profile [PP07]. The POLYAS system was checked against the security objectives defined in the PP in order to see whether changes are necessary or if the most current version will most likely pass the evaluation.

Almost all security objectives got a PASS (in total 23) during the evaluation whilst there are only two FAILs and six ORGs. Note, some of the PASSes were given even if in the current version the corresponding security objective is only ensured by organizational measures in the CC speech because they are based on the server's access control while the servers are not part of POLYAS. The FAIL from the protocol related security objective can easily be turned into a PASS. Also, the problem that the knowledge on the size of the vote message could possibly be used to link information has been discussed and a solution has been developed but has not yet been implemented. With respect to those security objectives that only received a PASS due to organizational assumptions, there are two possible steps that need to be taken: First of all, the system needs to know about the point in time when the election ends, and secondly the finite state diagram provided in the PP showing during which state what action exactly is allowed by the election commission needs to be implemented.

However, passing these security objectives is not a guarantee for passing the evaluation because during the evaluation the documentation and the development processes are checked as well. Moreover, the system needs to be tested by the evaluation lab.

Future plans include conducting the fourth election of the Gesellschaft für Informatik (GI) and the election of the Deutsche Forschungsgemeinschaft (DFG). In addition to that, next steps include the improvements of the few issues stressed in the analysis, looking for an evaluation lab and finally getting in contact with the Federal Office for Security in the Information Technology (BSI) in relation to the certification process. Hopefully, the evaluation will be finished in spring of 2008 and POLYAS will be certified for any further elections.

With this, POLYAS will bring a new level of transparency and security assessment to the world of remote electronic voting formerly only known from other areas of IT security.

References

[POLY07] Website of POLYAS (2007), http://www.polyas.de,retrieved2007-09-10

[KrVo05] Krimmer, R., Volkamer, M.: Bits or Paper? Comparing Remote Electronic Voting to Postal Voting. In: Wimmer, M.A., Traunmüller, R., Grönlund, Andersen, K.V. (eds.) EGOV 2005. LNCS, vol. 3591, pp. 225–232. Springer, Heidelberg (2005)

[GKMR06] Grimm, R., Krimmer, R., Meißner, N., Reinhard, K., Volkamer, M., Weinand, M.: Security Requirements for Non-Political Internet Voting. In: Krimmer, R. (ed.) Electronic Voting 2006, vol. P-87, pp. 203–212. Gesellschaft für Informatik, Bonn (2006)

[VoKr06] Volkamer, M., Krimmer, R.: Die Online-Wahl auf dem Weg zum Durchbruch. Informatik Spektrum 29(2), 98–113 (2006)

[PP07] Bundesamt für Sicherheit in der Informationstechnik, Schutzprofil Basissatz von Sicherheitsanforderungen an Onlinewahlprodukte, Bonn (2007)

[CC06] Common Criteria, Common Criteria for Information Technology Security Evaluation, Version 3.1 (2006)

Electronic Voting in Belgium: Past and Future

Danny De Cock and Bart Preneel

Dept. Electrical Engineering ESAT-COSIC, K.U.Leuven, Belgium
{Danny.DeCock,Bart.Preneel}@esat.kuleuven.be

Abstract. This paper provides an overview of the electronic and paper-based voting systems that are used in Belgium. It compares the advantages and disadvantages of these systems, and presents a selection of voting systems that have been recommended to the federal and regional governments for future elections in Belgium: an improved paper-based voting system with voter-verifiable paper trails, a family of optical scanning systems, a remote/Internet voting system, and a kiosk/intranet voting system.

1 Introduction

In December 2006, the federal and regional governments have requested a consortium of seven Belgian universities to deliver a study [1] on the technical, organizational, legal, and socio-political aspects of a range of voting schemes. In addition, special attention had to be paid to the accessibility and usability of the system for people with disabilities. The study was concluded in October 2007. In this article we report on the status of the current voting schemes in Belgium and outline the future solutions proposed in the second part of the study [2].

The article is organized as follows. Section 2 briefly describes the general principles of the elections in Belgium and describes the electronic voting scheme that is currently being used. Section 3 describes the recommendations that have been made to the administration. Conclusions are presented in Sect. 4.

2 Current Voting Schemes in Belgium

The Belgian state has a rather complex structure. The regionalization of the unitary state in the 1970s and 1980s has resulted in a three-tiered federation with federal, regional, and cultural/political community governments. This complexity is reflected in the structure of Belgian elections, where votes cast in one area can be transferred to others under certain circumstances. Moreover, the organization of the elections is also being regionalized, which means that the specific election details will start to vary between the regions (Flanders/Brussels/Wallonia).

The Belgian voting scheme using proportional representation with multiple parties (in practice more than 20). Voting is compulsory and the participation is in practice almost absolute. The voters are given the following options when voting:

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 76–87, 2007.

- vote for a list as a whole, thereby showing approval of the order established by the party they vote for;
- vote for one or more individual candidates belonging to one party, regardless of his/her ranking on the list (this has as effect that these candidates may end up higher in the ranking);
- vote invalid or blank so no one receives the vote.[1]

Votes for a list as a whole are distributed with a complex algorithm to the high ranked candidates on the list. Overall this description is a simplification, since each party also provides a list of substitutes, who take the place of elected politicians who for some reason decide not to take up their elected position. Ballot sheets may contain over 20 lists, each list may contain 30-40 candidates and 30-40 substitutes, so the overall sheet can be up to 1m×0.5m large. This size clearly presents specific logistics issues and may slow down counting of paper ballots.

Electronic voting was introduced with the federal elections in 1991. From that point on electronic voting was used in the local, provincial and European elections. In June 2007, 44% of the voters have used the electronic system. The main motivation for electronic voting is that the results are available faster. As the general population is drafted for the counting process (with a very modest hourly compensation), the reduction of long overnight counting efforts is an important argument. Other advantages are the reduction of costs, the elimination of spoiled votes[2] and a longer period to vote on election day.

In the electronic voting scheme, the voter receives an blank magnetic stripe card and goes to a voting booth. This booth contains a simple computer (80286 or 80386 processor) that has been booted from a floppy disk at the beginning of the election (no hard disk is present). The voter uses a light pen to select first the party and subsequently the candidates (and substitutes) for that party. When the vote is confirmed, it is written on the magnetic strip of the card; the voter takes the card and deposits it in the voting urn after the president of the voting office has confirmed that the card does not contain any marks that could identify the voter. This urn (that is driven by a computer similar to the voting computers) reads the magnetic strip and counts the vote; in the case of an audit or if a problem occurs with the urn's computer, a recount is possible by reading again all the magnetic stripes on a different machine. At the end of the election, the results are written to a floppy and brought by courier to the cantonal headquarters.

The system contains an extensive number of checks and balances. The voting booths and urn computers are started up following specific procedures (passwords and sealed floppies), test votes are issued at the beginning and end of each election and a voter can verify (but not correct) his vote in a different voting booth if he chooses to do so. All the software used is made available as

[1] It is impossible to cast an invalid vote with the the electronic voting system. Voters can, however, cast a blank vote.

[2] However, the impossibility to spoil a ballot may have increased votes for radical and protest parties.

source code to the political parties and is made public after the elections; note that two companies deliver a different version of the software (but every booth uses only one version).

A research team from the ULB checked for the legitimacy of electronic voting in 2003 [3]; the main conclusion was that 88% has a favorable attitude towards the system, while 8.5% are unfavorable (the remainder 3.5% had no opinion or did not answer). More detailed results are as follows: 95% of the voters find the system easy or very easy to use, 85% have no problem of principle with electronic voting, and 89% are fully confident or rather confident in the electronic voting system. A majority of voters who had confidence in electronic voting also expressed confidence in paper ballots, but in a more moderate fashion. On the other hand, the people who mistrusted the new method were those who favored mostly paper ballots.

While the Belgian electronic voting system is not as vulnerable as the DREs (Direct Recording Equipment) used in the countries such as the US (e.g., [4,5,6]), the Netherlands and France, a weak point is the computer of the voting urn: indeed, this computer may overwrite a vote and subsequent audit of the magnetic stripe cards will not reveal such an attack. Moreover, the central production and distribution of floppy disks requires a complex chain of custody in which the central point needs to be trusted (e.g., how can one be sure that the software made public later is the same as the software running on each voting computer?). Overall, very few problems have been reported so far; a major incident was the flipping of one bit in memory on the voting PC, resulting in an error of 4096 in the total count. In addition, some irregularities have been pointed out, for example, w.r.t. the handling of passwords to activate the devices. The voting machines themselves are "dumb" machines that do not store the ballots, but it is not inconceivable that one would modify the hardware of such a machine in order to attack the privacy and/or the integrity of the vote. These machines may also be vulnerable to side channel attacks, e.g., based on electromagnetic analysis. As these machines are rather large and not useful for any other purpose, it is not likely that they are stored in a location with high physical security. The paper based system is not without flaws either: the authors have been informed (off the record) that frauds are known in which votes to specific candidates are added during the counting process. It is also very easy to spoil a paper ballot by making an extra mark on it.

3 Recommendations for the Future Voting Schemes

As the electronic system is using hardware from the early 1990s, it is reaching the end of its lifetime. This, together with critiques from activists who do not trust that the content of the magstripe cards corresponds with the voter's choice, has prompted the Belgian administration to commission a study on the way forward. They have requested a team of seven Belgian universities to make concrete recommendations. This article focuses on the technical aspects of these systems, but it is clear that they need to fit in the broader Belgian context.

The following technical requirements must be met by the future electronic voting system:

Integrity: No election fraud; only eligible voters are able to vote; no one can vote more than once.

Transparency: Everyone must be able to verify that the election was conducted properly. In a relaxed form a number of independent experts must be able to verify that the election was conducted properly.

Privacy: No one learns how the voter has voted.

Vote Selling Resistance: Voter cannot prove how she voted (note that this is becoming increasingly difficult since a large part of the population possesses a mobile phone with a camera).

Incoercibility: A third party may not force someone to vote in a particular way – this property is a stronger version of the previous one as it involves an active opponent.

Usability: The vast majority of visually impaired or blind people should also be able to cast their vote autonomously.

After a detailed study of the Belgian voting schemes and some of the system used in foreign countries (including Estonia, France, Germany, Ireland, Latvia, the Netherlands, and Switzerland), four concrete options for the Belgian context have been developed:

1. An improved paper-based voting mechanism as an enhanced version of the current electronic voting scheme in which the magstripe is replaced by a barcode and a voter-verifiable paper trail.
2. An optical scanning mechanism using hand-marked voting ballots, perhaps enhanced with end-to-end security features for increased transparency.
3. A remote/Internet-based mechanism for use by Belgians abroad; it is similar to the systems currently in use for this purpose by France and the Netherlands. This Cybervote-style mechanism is a system based on homomorphic encryption [7,8]. The recommendation contains very explicit warnings w.r.t. the general insecurity of the user PCs (between 15 and 40% of general PCs are currently infected by spyware or have been part of a botnet, which presents a substantial security risk).
4. An intranet/kiosk-based mechanism which is essentially the same system as the Internet system, but as a PC in a controlled configuration is used, which makes it easier to manage the security risks.

Note that as scientists, we are fully aware of the risks related to the Internet and intranet schemes. However, the administration has requested that we explain and describe these risks; whether or not these risks can be accepted is a political decision.

The sections after the accessibility recommendations offer a detailed description of these options, together with an analysis to reveal whether they meet the technical requirements.

3.1 Accessibility Recommendations

As voting is a democratic right (and even an obligation in Belgium), the government has to guarantee that all citizens have adequate access to the voting process. The following accessibility guidelines have been put together based on a ministerial committee report of the Council of Europe [9], information from the GAMAH association [10], recommendations of the European Parliament's Disability Intergroup [11], the Belgian anti-discrimination act of 2003, and the UK Disabled Rights Commission [12]:

1. All authorities (and preferably also the political parties) involved in information distribution via the Internet should respect the Anysurfer guidelines [13] for accessible web page design. Anysurfer testing should be made compulsory for official websites related to the voting process. Attention must be paid to persons who need easy-to-read information.
2. Official websites must present an adapted simulation of the electronic voting procedure so that reading impaired persons can try out the procedures before going to the voting place.
3. Users shall be involved in the design of eVoting systems, particularly to identify constraints and to test ease of use at each stage of the development process.
4. Consideration shall be given, when developing new products, to their compatibility with existing ones, including those using technologies designed to help people with disabilities.
5. The electronic voting machines must have adapted outputs such as large characters and a synthetic voice (e.g., with a headset). In order to gain experience the development of experimental accessible voting machines and their testing should be stimulated.
6. When producing printed information material (flyers, brochures) related to the elections, authorities should make sure that different accessible formats are available for reading-impaired persons and other disadvantaged groups in the community.
7. Key access standards must not have the appearance of "optional extras": they must be considered core obligations; this should be reflected in any government procurement for voting equipment.
8. Physical access: voters with disabilities should have the choice to vote in voting places providing adequate access. On a longer time scale, administrations should strive to make all voting places accessible to all voters. Voters with disabilities must also be guaranteed the right to be accompanied in the voting booth by a person of their choice. Sufficient accessible parking places must be planned close to the voting places. Chairs must be available for persons who have to wait before casting their vote. The height of voting screens should be adapted for persons in a wheelchair or, better, should be adaptable.
9. Polling station personnel should be trained in disability awareness.
10. After every election, care should be taken to collect feedback from voters with disabilities in order to find out what should be improved for the next election.

3.2 Option 1 – Improved Paper-Based Voting

Description. The improved paper-based voting system allows the voter to inspect the paper ballot that is produced with a voting computer in a voting booth and to confirm that the voting ballot was correctly produced. The outcome of the elections is based on these paper ballots, which guarantees that the outcome effectively corresponds with the voters' choices.

Each voting office is equipped with a voting urn and a number of voting booths, and each voting booth hosts a voting computer equipped with a touch screen and a printer.

After successful identification of the voter as an eligible voter, the voter receives a voting token from the president of the voting office. A voting token can be used only once to make a voting computer produce a voting ballot. The voter uses the touch screen to mark the parties and candidates for which and whom (s)he wishes to cast a vote.

Once the voter confirms that (s)he has completed the voting process, the voting computer prints out the voting ballot. This ballot consists of two equivalent parts: an encrypted barcode and a human readable text with the voter's choice. If the voter confirms that the human readable text corresponds with his choice, (s)he folds the human readable part of the ballot to hide this information from unprivileged eyes. The barcode is encrypted using a public encryption key. The corresponding private decryption key is held by the election officials that have to process this municipality's voting ballots. The barcodes are decrypted at the Decryption Centers associated with one or more municipalities.

The voter presents the folded voting ballot to the president of the voting office to confirm that it does not contain any visible marks, after which the voter deposits the ballot in the voting urn.

At the end of election day, the urns of the voting offices of a municipality are assembled in disjoint sections, after which all the voting ballots of the urns of a section are mixed (this improves the privacy properties). Subsequently, the encrypted barcode of each ballot is scanned with a barcode reader. The list of these barcodes is then transmitted over an authentic channel to the Decryption Center to which the municipality is associated.

At the Decryption Center, the encrypted barcodes are decrypted to reveal the voters' plaintext choices. The sum of these choices result in partial results that are subsequently aggregated at municipality level, canton level, provincial level, etc. to eventually result in the election result.

Analysis. The proposed mechanism successfully increases the transparency of the voting process compared to the magstripe system, because the voting ballots that will be counted are those that have been verified by the voter. Furthermore, the voting computer does not contain any complex software that could be attractive to an attacker: it only produces a voter-verifiable paper trail (cf. the human readable text) that can easily and automatically be processed (cf. the barcode) to efficiently process the voting ballots.

If a large number of voters verifies and confirms that the choice they made with the voting computer corresponds with the human readable text printed

on the voting ballot, then it is trivial to audit the complete elections without having to care about the hardware used to cast the votes: independent auditors can check a random number of randomly chosen voting ballots and confirm for each of these ballots that the encrypted barcode indeed matches the human readable text.

Technical Requirements Checklist

Integrity: The voting token guarantees that one eligible voter is able to cast one vote.

Transparency: Under the condition that a sufficient number of voters verify whether the human readable text of their voting ballot effectively corresponds with their choice, independent auditors can conclude that the voting computers behaved correctly while producing the encrypted barcode, by verifying whether the encrypted barcode of a randomly selected set of voting ballots corresponds with the respective human readable text.

Privacy: The voter casts his/her vote in the privacy of a voting booth.

Usability: The visually impaired voter could use a braille reader and a set of ear phones to listen to the synthetic voice that informs the voter on the current stage of the voting process. The voting ballot, however, is not printed in braille, as this would compromise the voter's secrecy.

3.3 Option 2 – Family of Optical Scanning Voting Mechanisms

The second option consists of optical scanning mechanisms. The optical scan machines to record the votes require voters to mark their choices on well defined locations on a paper ballot. Optical scans are used in voting systems in order to increase the speed and accuracy of the voting process. The main advantage is that it can typically be implemented without any major modifications to the traditional voting procedure and that it can be subject to visual verifications.

Experiments of optical scanning systems have already been carried out in Belgium in the elections of 2003. They have turned out to be expensive and they have shown that ballots could not include more than twenty lists of candidates with a limited number of candidates per list. The system tested also required a prior manual verification of the ballots to exclude invalid ballots from the tallying.

The systems described below all share a number of common characteristics: in the voting booth, the voter marks the vote on the paper ballot by darkening one or more target areas on the paper ballot next to the identifiers or names of either lists and/or candidates; at some point in time, the paper ballot is scanned by an optical reader that recognizes which areas on the paper ballot were darkened by the voter and interprets this information to identify for which lists and/or candidates votes have been expressed. These systems do not use optical character recognition (OCR) techniques, as the voter marks his ballot by hand by darkening clearly specified target zones.

There are many choices to be made, which result in quite different systems, with specific advantages and drawbacks and which may induce specific requirements with respect to organization and procedures.

Types of Ballots. Different types of ballots can be used: (i) the size of traditional Belgian ballots depends on the number of lists and candidates, which makes them hard to handle, and it is nearly impossible to fold these ballots without folding the target areas that the voter can darken; (ii) fixed, standard size ballots are of a size for which scanning devices are readily available (A4, A3...). Because of the standard size of the ballot paper, rather inexpensive scanning devices can be used; (iii) double-sided, standard size ballots are appropriate for elections with many parties and candidates, as both sides of a sheet may be used to vote. However, there is a significant risk of forgetting to flip the ballot page when voting. This also complicates the scanning process, and requires more expensive scanning hardware; (iv) multiple sheet ballots consist of several (possibly double-sided) ballot sheets that accommodate a large number of parties and candidates, at the cost of a longer scanning process.

Since the folding of ballots may impair the scanning process, confidentiality of the vote must be ensured by other means. In the USA, "secrecy sleeves" or envelopes are used: the voter inserts his/her ballot in the sleeve or in the envelope before leaving the booth. The ballots must obviously be extracted before scanning. To prevent simple forms of ballot stuffing, ballots may be stamped before being handed over to voters.

Where and When to Scan. We examine four possibilities:

(i) In the voting booth, by the voter: the voter scans the ballot sheet(s) him/herself in the voting booth and the screen displays the scanned image and the vote set[3] which has been detected by the software. If the expressed vote is correct, and if the voter is satisfied that the scanning is accurate, he/she confirms the vote set. If the expressed vote set is incorrect, the vote set and the ballot are voided and the voter is asked to cast a new ballot.

(ii) In the voting office, with ballot verification: the ballot is scanned at the voting office in a setup which guarantees voter privacy and allows the same verifications as in the first case. The ballot is verified for conformity with election rules and the voter verifies that the scanning and vote set interpretation are accurate.

(iii) In the voting office, without ballot verification, the scanning is done at the voting office and the ballot is only verified for conformity with election rules. If the expressed vote is incorrect, the vote is voided and the voter is asked to request to cast a vote with another ballot.

(iv) The voting ballots are transported to decentralized scanning centers where the urns are emptied and mixed before scanning. Whenever a ballot is detected with an incorrect vote set, the vote set and the ballot are voided.

Technical Requirements Checklist

Integrity: The voter can only be sure that the recorded choice corresponds with his/her voting ballot if the voter scans the ballot. Organizational measures must enforce that an eligible voter is able to cast only one vote.

[3] A vote set is the collection of all votes expressed (marked) on a ballot.

Transparency: The voting procedure is very close to traditional paper ballot voting, which makes optical scanning easy to understand and accept. The paper ballot is inherently a voter-verifiable paper trail. Note, however, that the paper ballots may be counted manually if a recount is deemed necessary. There are effectively two independent ways to process votes: via the scanning and counting process and via a manual recount.

Privacy: The voter casts his/her vote in the privacy of a voting booth, but it may be necessary to enclose the voting ballot in a confidentiality sleeve to protect the voter's privacy.

Usability: Visually impaired voters cannot use this system to cast their vote.

3.4 Option 3 – Remote/Internet-Based Voting Mechanism

Description. The voting mechanisms in this section and the next one depend on homomorphic encryption. This means that the product of encrypted messages equals the encryption of the sum of the messages. This is extremely useful for electronic voting, where the encrypted message is the vote and hence there is no need to decrypt individual votes in order to get the final tally.

The remote voting system consists of server side components and client side components. The server is operated by election officials and contains mainly the voting server software, the election configuration manager software, a web server, and databases. The web server has two distinct functions: it displays election details and it allows for downloading remote voting and tallying client software. The voting server executes all the tasks related to server side voting functions. The election configuration manager reads the input provided by election officials, formats it and makes it available to other modules. Voters and (remote) talliers use fixed or mobile Internet devices, such as their personal computer, to connect to the remote voting servers for operations such as registration, voting and tallying.

The following players are involved in the elections. *Election officials* execute all the steps in the election process that need human intervention such as operating the computer systems and supplying the necessary data to the system. In any case, the election officials are responsible for the composition of the various ballot forms and the assignment of types of ballot forms to (classes of) voters. *Scrutineers* monitor the election process, both by verifying the actions of the election officials and by performing the necessary checks on the computer systems. *Administrators* are privileged users of the remote voting system who ensure the proper functioning of the computer and network equipment used during the whole voting process. Furthermore, the *Talliers* are appointed. These individuals will be responsible for determining the election result from the votes that were cast. The concrete tasks of the talliers depend heavily on the underlying (cryptographic) protocols. The number of talliers may vary form just one to tens or even hundreds. Depending on underlying (cryptographic) protocols, tallier keys are generated.

The remote voting process can be divided into three phases: the setup phase, the voting phase, and the tallying phase.

Setup Phase. During this phase the initialization of the election servers as well as the initialization of the organizational structure takes place. This involves the generation and configuration of all election parameters. All election officials, scrutineers and administrators are appointed and provided with the necessary credentials to correctly fulfill their obligations.

Voting Phase. This phase consists of voter authentication, ballot production, voting, and ballot validation. First, a voter authenticates him/herself to the remote voting server using the appropriate authentication mechanism issued to the voter prior to the election period. Belgian law requires that this authentication depends on the Belgian electronic identity card. Secondly, at the remote voting server, the authenticated voter's entitlement is checked, as well as his voting status to determine the correct set of elections. Based on this, a suitable ballot is created and sent to the voting client. The appropriate databases are updated to reflect this action. Next, given the ballot received from the remote voting server, the voting client software allows the voter to cast his/her vote(s) after which this software encrypts the vote. The voter digitally signs this encrypted vote using his/her electronic identity card. Eventually, the voting client software sends the signed and encrypted vote back to the voting server. Eventually, the remote voting server checks whether the digitally signed encrypted ballot corresponds with the voter who had been authenticated in the first step, after which the submitted vote is validated. If successful, it is stored in the Bulletin Board database. An acknowledgment is sent to the voter in any case.

Tallying Phase. At the end of the voting period, the votes need to be counted. The designated talliers use downloaded voting tallier client software to connect to the remote voting server through a secure connection over the Internet. The server tallying software processes all the encrypted votes in the Bulletin Board database to distribute chunks of encrypted votes to the talliers. Each tallier's tallying client software processes its chunk, decrypts its partial result, and publishes this sub-tally to the remote voting server. The tallying server software eventually aggregates all sub-tallies and publishes the election result.

Technical Requirements Checklist

Integrity: Remote electronic voting mechanisms depend on client-side software which is vulnerable to a multitude of security hazards: malware (viruses, Trojan horses, rootkits,...), OS version variations, bugs that are present in the OS and the application software. At the moment, it is impossible to guarantee an adequate protection level for voting from home.

Transparency: Only very few remote voting systems have been fully developed on paper, and almost none have been deployed in practice. Among experts, there appears to be a consensus that these systems still suffer from childhood diseases, and it is not certain whether the public is ready to accept fully computerized voting systems. Even though mathematical security proofs can be provided, they cannot be explained in simple terms to laymen. The absence

of a paper trail and the need to blindly trust the correctness of program code are often cited as reasons not to adopt systems of this type.

Privacy: The voter can cast his/her vote in the privacy of his/her office, at home, or at another location of his/her choice.

Usability: The visually impaired voter could use the computer system (s)he is familiar with to cast a vote using this system.

Coercion and vote buying: These risks are significant as it is impossible to prevent situations in which the voter casts a vote under pressure, or proves to a third party whom (s)he has voted for.

In spite of these weaknesses, it might be interesting to deploy this type of voting system for specific voters such as, for instance, Belgians residing abroad, as this would allow effective experimentation provided sufficient safeguards are implemented and might pave the way for greater automation of the voting process.

3.5 Option 4 – Intranet/Kiosk-Voting Voting Mechanism

Description. Kiosk voting refers to electronic voting systems in which voters must go to a voting office such as a polling station or an official building in order to register their vote on an electronic voting machine connected to a central server situated in a different location. Usually, all the votes are transferred to central servers which do the counting.

The distinction between Kiosk voting and remote voting is based on the fact that in a kiosk voting system, the traditional safeguards to guarantee the identity of the voter, the secrecy of the vote, and the free expression of the vote can be maintained. This kind of voting system does not have to face the typical concerns of remote voting, where the vote is cast in an unsupervised environment.

The implementation of a Kiosk voting system can also be based on homomorphic encryption. However, there is no need for the voter to authenticate himself to the voting computer. Instead, the voter physically authenticates himself to the president of the voting office, and receives from this president a voting token, with which he can activate the voting computer. The client software and configuration files to be used in voting computers are downloaded beforehand, as the voting computer will present the same lists to all voters.

Technical Requirements Checklist. Compared to the remote voting system presented in the previous section, the coercion and vote buying problems disappear as the voter can cast a vote in the privacy of a voting booth. However, many problems which are typical for networked applications remain.

The most important plus of a kiosk based system consists of the increased privacy provided by voting booth and the controlled environment which effectively deals with the issues related to the (in-)security of client-side operating systems and applications, as the computer systems used at the client side can be better protected against malware, operating system variants, etc.

4 Conclusions

Even if our study is to some extent specific for the Belgian context, we believe that it will be very helpful in evaluating and comparing a range of electronic voting systems.

We believe that it is possible to improve substantially over the current DRE-based electronic voting systems, by opting for a design as proposed with the improved paper-based voting mechanism that minimizes complexity and relies as much as possible on a paper trail that can be verified by the user and that provides a means to audit the elections rather than the election equipment.

We are more cautious w.r.t. network-based systems. We believe that the current systems may not be fully ready for prime time; however, there are important research challenges in this area and experiments with small systems should definitely be encouraged.

References

1. Consortium of Belgian Universities, BeVoting Study of Electronic Voting Systems, Study (April 15, 2007)
2. Consortium of Belgian Universities, BeVoting Study of Electronic Voting Systems, Recommendations (October 12, 2007)
3. Delwit, P., Kulahci, E., Pilet, J.-B.: Le vote électronique: un choix légitime? In: Bruxelles/Gent: Politique scientifique fédérale, Academia Press, San Diego (2004)
4. Bishop, M.: Overview of Red Team Reports, http://www.sos.ca.gov/elections/voting_systems/ttbr/red_overview.pdf
5. California Secretary of State, Top to Bottom Review (2007), http://www.sos.ca.gov/elections/elections vsr_htm
6. Wagner, D.: Principal Investigators Statement on Protection of Security-Sensitive Information (August 2, 2007), http://www.sos.ca.gov/elections/voting_systems/ttbr/State_of_protect(DW).pdf
7. Cramer, R., Gennaro, R., Schoenmakers, B.: A Secure and Optimally Efficient Multi-Authority Election Scheme. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 103–118. Springer, Heidelberg (1997)
8. EU IST Project Cybervote (2000-2003), http://www.eucybervote.org
9. http://www.cev.ie/htm/report/second_report/pdf/Appendix
10. Vandecandelaere, M.-A., Pierre, J.: Electeurs à mobilité réduite: Citoyens à part entière ou entièrement à part? Le Vilain Petit Canard, 4–7 (March 2006), http://www.gamah.be/pdf/VPC_MARS2006.pdf
11. European Disability Forum, http://www.edf-feph.org/Page_Generale.asp?DocID=8863
12. Disability Rights Commission, http://83.137.212.42/sitearchive/drc_gb/easyread/votingrights/index.asp.html
13. http://www.anysurfer.be/
14. Council of Europe, Committee of Ministers, Recommendation Rec, 11 of the Committee of Ministers to member states on legal, operational and technical standards for e-voting, Adopted by the Committee of Ministers on 30 September 2004 at the 898th meeting of the Ministers' Deputies (2004), http://www.coe.int/t/e/integrated_projects/democracy/02 activities/02_e-voting/01_recommendation/Rec(2004)11_Eng_Evoting_and_Expl_Memo.pdf

The Digital Voting Pen at the Hamburg Elections 2008: Electronic Voting Closest to Conventional Voting

Joerg Arzt-Mergemeier, Willi Beiss, and Thomas Steffens

Freie und Hansestadt Hamburg, Landeswahlamt,
Johanniswall 4, 20095 Hamburg, Germany
landeswahlamt@bfi-a.hamburg.de

Abstract. Due to recent changes of the election law of the Free and Hanseatic City of Hamburg, Germany, the counting of the votes at the next elections for the state parliament will be complicated and time consuming. To nevertheless enable the Election Supervisor to announce the preliminary results of the election on the evening of the Election Day the Parliament has chosen to make use of an electronic voting system, i.e. the Digital Voting Pen System ("Digitales Wahlstift-System" - DWS). The main reasons for favoring this electronic voting system have been its closeness to the conventional voting procedure and therefore its acceptance among the voters, its security, and its verifiability.

1 Introduction

Recent legislative actions taken by the people of the Free and Hanseatic City of Hamburg (City of Hamburg) as well as the state parliament, the "Hamburgische Buergerschaft", regarding the election law have major practical effects on the next parliamentary elections in Hamburg on February 24, 2008. Part of the new and rather complicated election law is that each voter has more than one vote and can either cumulate its votes or cross vote, i.e. split its votes for different candidates and parties, or even use a combination of both on its ballot. Furthermore, due to the number of candidates a ballot booklet instead of a single sheet ballot paper will be introduced, which the voter has to open and leaf through. Depending on the electoral district the authorities expect ballot booklets up to fifteen pages. Considering the 1.2 millions people entitled to vote in Hamburg and the fact that these ballot booklets are complicated and time consuming to count, it would be impossible for the responsible Election Supervisor to announce as usual the preliminary results of the elections on the evening of the Election Day, unless the votes will be cast with technical support. Without technical support a period of about three days was regarded as necessary in sample calculations considering a reasonable number of counting clerks [2].

For this reason the State of Hamburg parliament has decided to introduce an electronic voting device based on a digital pen technology for the elections taking place in Hamburg in February 2008. Among other reasons in doing so the

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 88–98, 2007.

aim was to keep the voting procedure itself to a very high degree unchanged, i.e. the votes should still be cast in the familiar way by using pen and paper. This should enable the voters to concentrate on the new possibilities of choosing candidates as provided by the new election law instead of having to learn a new election procedure or even to bother with unfamiliar technologies. Hereby the parliament took into account the fact that a real digital voting pen system had not yet been developed or even approved by the relevant authorities.

The acceptance of a digital pen solution was tested in Hamburg in the course of the German federal elections in 2005. 677 voters took the chance to test a standard digital pen as a voting device. Afterwards 504 of them participated in an enquiry. 84 % of them stated that they would welcome the use of a digital pen for elections. From the remaining 16 % many felt not sufficiently informed about the use of the pen, which may give reason to expect an even higher degree of acceptance with an adequate information policy ahead of the elections [4].

To develop the Digital Voting Pen System ("Digitales Wahlstift-System" - DWS) in close collaboration with the City of Hamburg a consortium of two German companies, Diagramm Halbach and WRS-Softwareentwicklung, was chosen in a European-wide procurement procedure which took place in 2006. The consortium has recently completed the development, programming and setting up of the DWS. Presently the system is undergoing different certification procedures.

This paper contains a brief summary of the concept of the DWS. It describes the reason for the introduction of this new voting device as well as some technical and legal aspects connected to it.

2 Functioning of the DWS: The Anoto Technology

The DWS is based on a digital pen and paper solution using the so-called "Anoto technology". This technology has been developed by the Swedish enterprise Anoto Group AB in Lund[1].

A digital pen can be used like an ordinary pen, i.e. it can be used for writing or drawing on regular paper. It differs from a common pen in so far as it has also included a little camera plus microprocessor which scans the marks the user places on the paper. For taking advantage of the digital pen technology, besides the digital pen one generally needs specifically prepared paper and a corresponding software application. The paper must contain the printing of a specific (patent-protected) dot pattern in the background, which is barely visible. In the case of the DWS this pattern is unique and must be different on every page of a ballot booklet. The pattern is scanned by the camera once the pen touches the paper and indicates the exact position of the digital pen. The coordinates of the marks which are made on the paper with the pen are stored in the pen, then - in the case of the Hamburg elections - transferred via a docking station and a USB-cable to a stand-alone notebook and can be further processed there for the counting procedure (see below). Although standard digital pens contain devices for wireless communication these are regarded as not secure for the use as a

[1] See http://www.anoto.com/

voting device; therefore they are deactivated, and instead the wired connection via docking station is preferred.

Following DWS-equipment is required for each of the about 1.300 polling stations: A computer (notebook) with the specifically designed election software, a portable storage device (USB flash drive), three docking stations for the digital pen plus an USB hub, a number of digital pens and a battery-charging station in case a pen's battery is empty, a printer for hardcopy documentation of the election procedure, and the ballot booklets (cf. Figure 1).

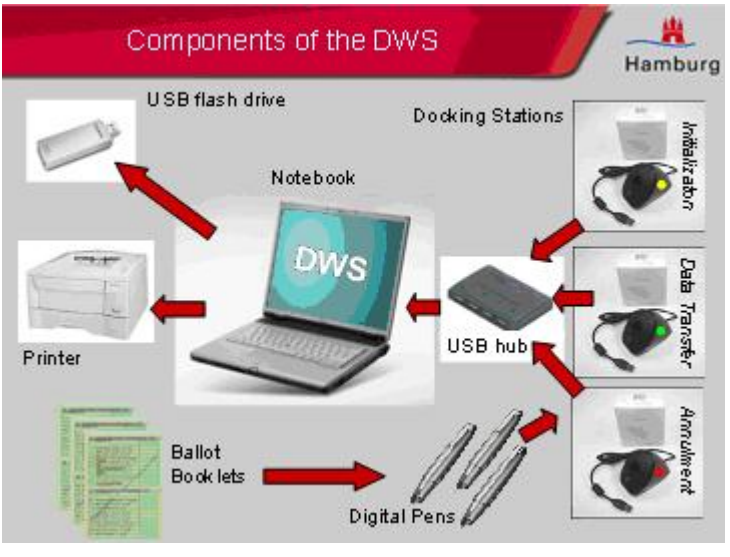

Fig. 1. Components of the DWS at the Hamburg Elections 2008

3 Use of the DWS on Election Day

The most specific advantage of the DWS is that the voting procedure remains to a very high degree unchanged for the voter: The voter will cast its vote by using a pen and paper, maybe noticing only the bigger size of the pen, possible vibrations of the pen when signalizing activities, and the slightly grey pattern on the ballot booklet. The electoral officials though have to be thoroughly instructed in the use of the DWS.

3.1 Voting Procedure from the Voter's Point of View

When arriving at the polling station the voter has to sign in, i.e. it has to show its voting card. An electoral official checks the voting card and hands over a Digital Voting Pen to the voter. Besides the pen the voter receives the ballot booklet (as a matter of fact at the next Hamburg election the voter will receive

four different kinds of booklets due to two different elections taking place at the same time with two booklets for each election). The voter then enters the polling booth and sets its marks with the digital pen. In difference to former elections the voter must not use another pen of any kind to mark its vote otherwise the ballot will be counted as invalid.

The voter then returns to the officials and sticks the digital pen in a docking station. On the computer screen, which is turned to the voter, an animation is shown to the voter which indicates that its vote is transferred to the computer and registered. Of course it is neither shown which data are transferred or even whether the voter has marked anything on the paper ballot or not, nor can this be deduced from the time needed for the data transfer. If the voter has - intentionally or by mistake - used a digital pen which has not been registered for this polling station or has not been activated correctly, an error notice will appear on the computer screen. As a final step after the vote is registered in the computer the voter casts its paper ballot booklet into the ballot box and may leave the polling station.

If the voter wants to correct its vote after leaving the polling booth but before transferring the data to the computer it can still do so by telling the electoral officials. The voter then sticks the digital pen in a third docking station to empty it without transferring and storing its data. The voter then has to destroy the ballot booklet (watched by an electoral official) and may restart the voting procedure from the beginning.

To inform the voters about the use of the DWS ahead of the elections different actions are taken: In the course of a broad information campaign which is intended to inform the voters about the new election law also the introduction of the DWS will be spread, due to the complexity of the topic mainly by public relations measures and via internet. In addition to that the election authorities in Hamburg will offer "sample polling stations" to experience the use of the DWS. The premiere of this will take place on September 8, 2007, the Open Day of the state parliament, and then be continued in November 2007. The sample polling stations will be set up in shopping centers or similar venues. Besides general information on the DWS given there, in the sample polling stations the visitors will be offered the chance to watch how a sample version of the DWS works and to participate in sample elections.

3.2 Handling by the Electoral Officials and Data Processing

The Election Day for the scrutineers and other electoral officials starts differently from former elections. The distribution of the DWS as well as the other equipment needed for the conduction of the election is done early in the morning following a strict and secure logistic concept including the storage of the DWS itself in locked and sealed boxes and its transport by trained personnel. The distribution must be finished by 7:30 am at the latest by handing over the equipment to the local electoral officials.

First thing in the morning the electoral official has to set up the hard- and software, as it is trained in advance To do so it must turn on the computer and

is then guided through the installation procedure by the software which starts automatically. Additionally the official is supplied with a handbook to accompany the installation procedure and possible incidents occurring throughout the day, and is also offered hotline support. The docking stations and printer must be connected to the computer, the identification of the polling station must be entered and the digital pens which are specifically set up and registered for the Hamburg elections, must additionally be registered digitally for this specific polling station. From that time on the digital pens can be used in no other polling station but the one they are registered for. Neither can standard digital pens be used for the election. The correct installation of the proper DWS is verified by secure hashes.

When a voter wants to vote the electoral official checks if it is entitled to do so, initializes one of the digital pens in the first docking station, tests the operational reliability of the digital pen, and hands it over to the voter. When the voter returns to the electoral official from the polling booth, the coordinates of the voter's marks are stored in the pen. The electoral official assists the voter in sticking the pen in the correct docking station either to register the vote in the second docking station or to annul the vote by emptying the pen in the third docking station. If the vote is ready to be registered the coordinates of the voter's marks will be transferred to the computer via the USB cable and randomly stored on the computer's hard disk as well as on a separate USB flash drive. The pen is then automatically emptied and deactivated.

When the polling station closes at 6:00 pm the scrutineers proceed to the next step by starting the evaluation of the votes. Hereby the stored coordinates are matched with an electronic version of the ballot booklet. The software validates each ballot by applying rules for separating clearly valid from clearly invalid ballots. These rules are subject to the certification procedures by the national metrology institute and the Federal Office for Information Security (see sec. 5 below). Those ballots that cannot be classified by the software as clearly valid or clearly invalid must be checked manually (to be more precise: on screen) by the responsible electoral official, who decides on the validity of these votes. If desired though by the electoral officials they can also check the clearly valid or invalid ballots on screen as well. Once all ballots are validated the software calculates the result of that specific polling station, stores it, and prints it out. The scrutineer then takes the USB flash drive and the signed print out to the next main administration building which is equipped with a secure wired data connection and transfers the data to the central data processing center. An authentication to ensure the USB flash drive has not been subject to manipulation during transport is finally done again by secure hashes.

3.3 Postal Voting

Obviously the DWS cannot be used as a voting device for postal voting. The electoral officials therefore still face the problems connected with the complicated and time consuming task of the manual counting of votes. For this reason it is

intended to use the DWS as a counting device for the postal votes to support the electoral officials on the evening of the Election Day.

4 Legal Issues

The introduction of the DWS requires several activities with regard to the electoral rules and regulations. In the first place the use of the Digital Voting Pen System must be approved by the competent authorities of the City of Hamburg. Secondly the details of the use must be set up in the state election law, particularly the monitoring of the DWS on Election Day. For example regulations must stipulate the procedure in the - unlikely - event of malfunctioning or other incidents of loss of relevant data.

Furthermore in Hamburg it is decided - and must therefore be clearly provided for in the election law - that the DWS is a voting device instead of a plain counting device. As a result the electronic vote is the relevant vote. A counting of all the paper ballot booklets is neither required nor intended (apart from control samples, see sec. 5.5 below). This decision may also be important for example in the case a voter uses another pen instead of the digital pen to mark its ballot booklet. In this case the paper ballot would show a cross whereas the digital pen would have not recorded a mark, and the DWS would therefore classify the ballot as invalid.

5 Technical and Security Issues

In order to present a reliable and secure electronic voting device the DWS is undergoing several testing and certification procedures by different institutions before it will be approved as a voting device by the state Department of Interior [1]. These procedures are currently taking place.

5.1 Objectives

The main objectives for the security standard can be derived from the basic principles for democratic elections, i.e. elections have to be universal, direct, free, equal and secret (as provided for e.g. in the German constitution, the "Grundgesetz" (Basic Law)) [3]. The resulting objectives can be summarized as follows:

- The identity of the voter must not be linked in any way to the vote it has cast.
- Every voter must have the chance to cast its vote, but it is allowed to cast its vote only once, and each vote must be counted only once.
- There must not be any possibility to change or delete votes once they are cast, or to add votes which had not been cast in a regular way.
- There must be no possibility to get intermediate results before the closing of the polling station.
- The results must be counted correctly and completely.

To meet the above mentioned objectives various functioning and security concepts have been developed and are the basis for the examination by the institutes involved. The following measures required herein shall be pointed out in advance:

- The votes are stored anonymously and in a randomized order. Neither the time nor the order of the casting of votes can be traced. Therefore identity and identification are not critical issues when using the DWS.
- There will be no wireless data transfer (for example via Bluetooth) involved: Any devices in the notebooks or the digital pens enabling wireless data transfer are disabled. Instead the data are transferred from the digital pen to the computer via a USB cable connection. Similarly the final transfer of the results to the central data processing centre is done via cable connections using the City of Hamburg data infrastructure (cf. Figure 2).
- The data stored in the digital pen are erased once they are transferred to the computer.
- There exists no function in the DWS which would enable the electoral officials to check the contents of a vote, i.e. neither the place or type of marks nor their number can be detected by the officials. The computer signalizes only that a vote has been cast. It does not evaluate or classify the vote as valid or invalid until the end of the day.
- The digital pens contain another than the standard firmware specially developed for election purposes. This includes the deactivation of different features such as wireless communication (Bluetooth) or update capabilities. A manipulation of the DWS via the digital pens is regarded as not possible.
- The computers (notebooks) are configured in a way that all features which are not needed for the election are deactivated as well (DVD, WLAN, Bluetooth, infrared, seriell and parallel ports). Furthermore the BIOS contains various security settings.

5.2 Certification by the National Metrology Institute (PTB)

One institution to certify the DWS is the Physikalisch-Technische Bundesanstalt (PTB). The PTB is the national metrology institute performing fundamental research and development work in the field of metrology as a basis for all the tasks entrusted to it. Among other tasks the PTB is responsible for type approvals.

For the type approval of the DWS the PTB will mainly focus on the evaluation of functional requirements of all software and hardware components. The evaluation will be based on the “Regulation for the Use of the Digital Voting Pen System at Elections for the State of Hamburg Parliament and for the Assemblies of Local Citizens as well as at Referenda”, a set of requirements especially developed for the DWS by the City of Hamburg in close collaboration with the PTB.

5.3 Certification by the Federal Office for Information Security (BSI)

Due to its largely software relied approach the DWS is also certified by the Federal Office for Information Security (BSI). The BSI is the central IT security

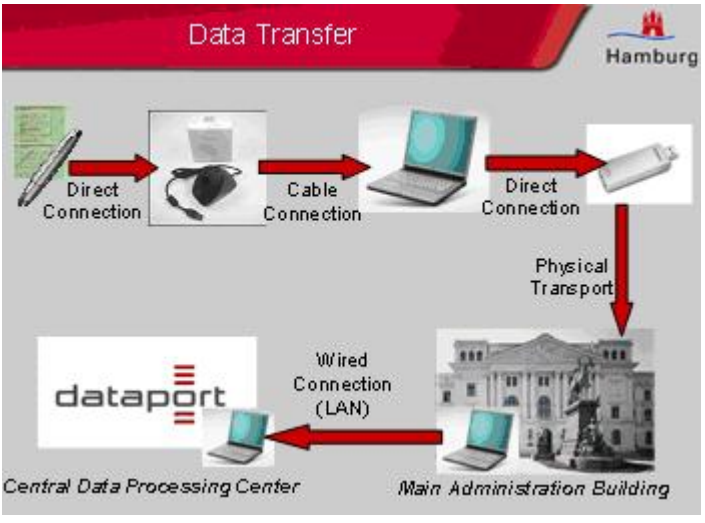

Fig. 2. Data Transfer from the Digital Pen to the Central Data Processing Center

service provider for the German government. As Germany's National Security Agency, it is its goal to promote IT security in Germany.

The BSI is certifying the DWS on the basis of the Common Criteria for Information Technology Security Evaluation (Common Criteria), an international standard for computer security. The certification is following a two-step approach: In a first step the German Research Center for Artificial Intelligence (DFKI) on behalf of the City of Hamburg has developed a Protection Profile which applies to the technical assistance for elections in the polling station (Protection Profile BSI-PP-0031-2007 [5]). It describes the requirements for IT-security of technical systems for assistance in elections, based upon the usage of digital voting pens. These security requirements are based on the intended usage assumptions, threats and objectives. They concern the digital pens and their firmware, the docking stations, and the election evaluation-software. The hardware (notebooks by Fujitsu Siemens Computers) as well as the system software (Microsoft Windows XP SP2) are not part of the evaluation itself but are accounted for in the requirements for a secure environment. This Protection Profile has been certified on March 14, 2007. In a second step the DWS-system itself is now undergoing the certification procedure by the BSI whether it complies with this specific Protection Profile.

5.4 Further Certifications

The DWS is undergoing further examination with regard to its electromagnetic compatibility, its robustness, and finally relevant data protection issues. These examinations are done by accredited partners.

5.5 Organizational Measures to Assure Security

Apart from the technical measures to provide a secure solution for an electronic voting device further actions must be taken to assure the correct conduct of the Hamburg elections. Among the broad variety of measures taken the following shall be expressly mentioned:

- The electoral office's information policy in the run up to the elections contains a variety of measures to provide for transparency and information to the public about the DWS and its use. This includes detailed information via internet and handouts for example, as well as events where sample DWS-systems can be viewed and tested by the public.
- Already in the procurement concept the manufacturers of the DWS have been required to deliver an easy to use and easy to explain solution, which can be handled by most users intuitively.
- Detailed introduction of the DWS, training and information for the scrutineers and other electoral officials keeping in mind the possibility that the majority of them have litte (or even no) experience with computers.
- Before handing the digital pen to the voter the electoral official in the polling station can easily check the correct functioning of the digital pen, i.e. by a simple testing procedure it can confirm that the pen has not been manipulated.
- The data are transferred from the digital pen to a stand-alone notebook, therefore no threat is posed on the DWS by any internet security risks. The transfer is done via a secure connection by USB cable, therefore no threat is posed on the DWS by any security risks due to wireless connections such as Bluetooth, infrared etc.
- The fact that the DWS is paper-based is used for its verfiability: The paper ballot booklets are not only used as a Voter Verified Paper Audit Trail (VVPAT), but they are also used for taking control samples after the closing of the polling stations solely with the intent to examine and verify the correct functioning of the DWS. This will be conducted by comparing the electronic results with the results of the correspondent paper ballots in 17 polling stations being chosen at random. The 17 out of about 1300 polling stations (1.3 %) represent each of the 17 electoral districts.

Finally the overall logistic concept plays a major role in ensuring a reliable and secure use of the DWS. This concept must for example consider apparently trivial aspects like weather conditions on Election Day: Computers must not be stored below a certain temperature before being used in well-tempered rooms. Similarly important though is the organization of the transport of the different components from the manufacturers to the City of Hamburg in the run-up to the election and especially on Election Day. All these transports are classified with a high demand for security which requires for example the transport in locked and sealed boxes or trucks, and other means such as an escort.

6 Cost

The costs of the use of the DWS total about EUR 5 millions. This includes the development of the DWS, its programming, purchase of the digital pens, transport and further logistic costs, and the costs for the various certification procedures. This sum does not include the costs for the printing of the ballot booklets and the costs for the computers being used in the polling stations. The latter will be purchased from the general computer supplier of the City of Hamburg and will be used to satisfy any replacement demand in computers by the City of Hamburg after the election is finished.

Considering that parts of the costs involved are non-recurrent expenses, i.e. expenditures for the development and programming of the device itself, its certifications, the digital pens etc, the DWS is regarded a cost-saving solution especially if used again at further elections. To assess the relevance of these costs it must be looked at in proportion to the potential costs of conventional counting of votes. Because of the complexity and number of ballot booklets due to the changes of the election law these would amount to about EUR 7 to 8 millions. They would mainly be caused by the personnel to perform this time consuming task.

7 Conclusion and Perspectives

The use of an electronic voting device at the next Hamburg elections is not an end in itself. Technical support is rather necessary due to the new election law featuring a multiple vote system which enables voters to cumulate or cross votes. The DWS is regarded as a convenient and secure solution for supporting elections. The main advantages of the Digital Voting Pen compared with e-voting solutions having so far been used in other German or international elections is seen in the fact that the voting procedure remains almost unchanged and that the voter receives a reliable receipt of its vote in the form of the paper ballot booklet. This booklet is collected parallel to the electronic vote and will be used for spot tests or in depth verification of the results of one or more polling stations, or - in the worst and unlikely case that the DWS malfunctions on Election Day - can be relied on as a substitute ballot. This possibility of proving the results of the electronic voting device goes beyond the demand for a Voter Verified Paper Audit Trail (VVPAT). The lack of such audit trails has been widely criticized at other elections where e-voting devices had been used.

It is suggested that different aspects beyond the technical and organizational measures should be observed in the aftermath of elections by the competent research authorities. Such aspects include the social and political effects of the use of electronic voting devices such as the Digital Voting Pen as well as its impact on the voters' behavior. These effects can be regarded as difficult to predict and to evaluate. Nevertheless they should not be left out in the various election analyses. Furthermore the use of electronic voting devices should not be discussed without keeping an eye on the relevant election law and their interdependency. The elections in Hamburg will be the world premiere of an electronic voting

device based on the digital pen technology. Though it is said that a "smart pen" has for the first time been used in elections by the Scottish Community of Clackmannanshire at its council election[2] in September 2006 there remains one important difference to the Hamburg elections: In the Scottish election the manually counted result was compared with the data captured through the digital pens during the electronic voting trial. Legally the results of the traditional paper ballot formed the basis of the election result. Therefore in Scotland the digital pen was merely used as an "electronic counting device". On the contrary in Hamburg the results of the electronically cast vote form the basis of the election result which makes the Digital Voting Pen a true "electronic voting device". When its reliability has been confirmed in numerous tests the DWS in February 2008 still has to prove in real practice of an election of this kind with 1.2 million voters that it is capable to meet all its strict requirements. Due to the responsibility that arises hereof the City of Hamburg aims at maximum transparency (as long as it is compatible with necessary security aspects), and welcomes the lively and differentiated discussion process of national and international experts, supporters and critics, dealing with electronic voting devices.

References

1. Gesetzüber die Wahl zur hamburgischen Bürgerschaft (Bürgerschaftswahlgesetz) (2007)
2. Manfred Jäger. Unterstützung der Auszählung der Wahl zur Hamburgischen Bürgerschaft und den Bezirksversammlungen durch technische Hilfsmittel. Technical Report Drucksache 18/3569 (2006)
3. Rat, P.: Grundgesetz (GG) für die Bundesrepublik Deutschland (retrieved on 15-12-2005, 26.07.2002), http://www.datenschutz-berlin.de/recht/de/gg
4. Freie und Hansestadt Hamburg. Pilotstudie zum Digitalen Wahlstift. Technical Report (2005), http://fhh.hamburg.de/stadt/Aktuell/wahl/digitaler_20wahlstift/start.html#headline5
5. Volkamer, M., Vogt, R.: Digitales wahlstift-system. Common Criteria Protection Profile BSI-PP-0031, Bundesamt fr Sicherheit in der Informationstechnik (2006)

[2] See http://news.bbc.co.uk/2/hi/uk_news/scotland/tayside_and_central/5385086.stm

The Security Analysis of e-Voting in Japan

Hiroki Hisamitsu and Keiji Takeda

Carnegie Mellon CyLab Japan
1-3-3 Higashikawasaki-cho, Chuo-ku, Kobe city
Hyogo prefecture, 650-0044 Japan
hhisamit@andrew.cmu.edu, tkeiji@cmu.edu

Abstract. To assess trustworthiness of e-voting practices in Japan, security of e-voting systems and their operational procedures are analyzed. All e-voting systems available on the market are covered in the analysis. Through the analysis we concluded that current e-voting security is heavily depending on protection by operational process rather than security features of the system and it is confirmed that the systems provide only limited security features and there is large room for technical improvement. Typical security issues are lack of protection mechanism of programs and data on tabulation machines. This vulnerability enables malicious poll worker or manufacturer to insert malicious code to generate arbitrary election results.

Keywords: e-voting, e-voting system, e-voting operation.

Background

E-voting, an election method that allows voters to directory record electronic voting data on an electronic media, is spreading in various countries around the world. In general e-voting is thought to contribute to a higher turnout rate in elections, swift tabulation, cost reduction, and better accessibility for disabled voters. Countries such as the U.S., UK, Brazil, and India have already deployed e-voting widely [6, 9, 10, 12, 14, 16]. Korea and Estonia are applying more advanced methods of Internet voting. However, various problems such as technical failure, security problems, etc. have already been pointed out in real elections. Many researchers and the media have repeated security issues around e-voting [4, 5].

In Japan, only local elections are allowed to use e-voting by law at this time. The discussion of introducing e-voting into national elections in the Diet was just initiated in 2007. There is a possibility that adoption of e-voting will be approved by law in 2008.

1 Introduction

1.1 Objective

To assess trustworthiness of e-voting in Japan, we investigated the reliability and security of e-voting systems and their operational procedures by election officials in Japan, and analyzed them from the viewpoint of information security. The practice of e-voting in

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 99–110, 2007.

Japan is still at the initial stages and this research would be the first trial to comprehensively assess e-voting in Japan.

There have been many research projects on e-voting systems conducted especially in the U.S. [13]. We recognized that those researches are mostly to identify security flaws on hardware and/or software of e-voting machines. The e-voting system consists of e-voting machine, registration machine and tabulation machine. The researches of three types of machines are required to establish secure e-voting systems effectively. Although such research contribute to establish secure e-voting, in reality security of e-voting can be kept by both security features of e-voting systems and the operational procedures or management of the election by the officials. We have not found existing researches that analyze both systems and operational procedures for information security of e-voting. To bridge these gaps, we have researched the security and reliability from both aspects of e-voting systems technology and the operational procedures of elections with e-voting [15].

1.2 Scope

In this research, we examined all available e-voting systems and examined e-voting operational procedures conducted by election officials. Secure e-voting practices can be established as combination of following factors. The first is "technology" (e-voting system) which ensures accuracy, robustness and verifiability of voting data. The second is "policy" which administrates the principle of e-voting, this includes laws, regulations and guidelines. The last factor is "operational procedure" e.g. election operation, manual verification, physical protection, this ensures security of e-voting by human operation or management. Past security analysis of e-voting in the world has been conducted from only the technological aspect. However, our approach is aiming to be comprehensive. (See Figure 1).

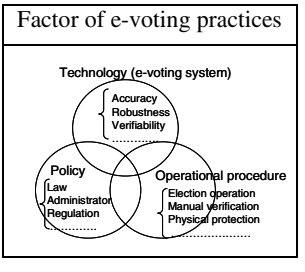

Fig. 1. Three security factors of e-voting practices

1.3 Related Works

There has been research that analyzed security of e-voting systems [7,13]. Most of these security researches are analyzing hardware or software of the e-voting system itself and ended with vulnerabilities found on the systems [8]. Regarding the Kohno's report [2], they found some flaws where they describes "most notably, voters can easily program their own smartcard to simulate the behavior of valid smartcards used in the election." Currently e-voting manufacturers adopted security mechanism every

election. Then their product utilizes random numbers assigns for every IC cards at the beginning of the election. It makes difficult for attackers to counterfeit IC cards. Therefore, the attacks pointed out in this case are not applicable for modern e-voting systems. Other research mentions as "undesirable modifications could be made by malevolent poll workers (or janitorial staff) with access to the voting terminals before the start of an election" [2]. "An attacker who gets physical access to a machine or its removable memory card for as little as one minute could install malicious code; malicious code on a machine could steal votes undetectably, modifying all records, logs, and counters to be consistent with the fraudulent vote count it creates. An attacker could also create malicious code that spreads automatically and silently from machine to machine during normal election activities—a voting-machine virus" [3]. The research is not realistic because physical accesses to e-voting systems are usually strictly prohibited and it is hard even for malevolent poll workers and janitorial staff to gain access to attack. For example, all e-voting systems are delivered by manufacturers and are kept in a depository with a security guard in Japan. Ariel's study also presented a video movie on how to pick a lock on the cover of e-voting machine in 10 seconds. Though technically this is possible however this is not feasible in real operational procedures situation since there are many security guards and election officials watching those machines. Ariel's study ignored such physical or manual protection of the machines. Therefore, we should evaluate both election official management procedures and e-voting systems.

2 Analysis of e-Voting in Japan

Japan's law has permitted e-voting at local elections since 2002. Japan does not have much e-voting experience, though currently the Japan's government is promoting e-voting adoptation. Only 9 among 1800 local governments conducted their elections via e-voting. There have been troubles on e-voting applications [11]. One of the elections became invalid because of discrepancies between the total number of voters and the total number of votes. Five local governments have withdrawn from using e-voting. Some e-voting machine manufacturers who supplied systems for this election have withdrawn from the market. However, the government plans to pass a law that introduces e-voting into the national elections in the coming years.

2.1 Laws in Japan

The Ministry of Internal Affairs and Communications (MIC) is in charge of elections in Japan. All elections are implemented based on The Public Office Election Law, the e-voting special law and local government ordinances. National elections are conducted only through paper ballots and the local elections are conducted through either paper ballots or e-voting. Whether the election is held by paper ballot or e-voting is decided by local governments. Japan's government passed the e-voting special Law in 2002. Also, the absentee vote by e-voting and paper ballot were permitted in 2004. The following is a summary of each law. The Public Office Election Law defined the scope of candidate, polling method, date, time, tabulation method, etc. for elections of local congressmen and

Mayors. The e-voting special Law was introduced in Feb.1, 2002, permitting e-voting systems to be used for local elections of local council and mayors. It defined e-voting machine as follows.

> "The method in which voters go to the polling station by themselves, operate the electronic voting machine, choose only one candidate from the list, and that vote is stored on an electronic media" [1].

2.2 History of e-Voting in Japan

The first e-voting election in Japan was conducted in 2002 at Niimi city of Okayama prefecture. This election ended successfully. All together, 9 local governments have conducted 16 e-voting elections so far. The biggest e-voting election was Yokkaichi city in Mie prefecture and the turnout was approximately 95,000 people. The smallest voter turnout was around 5,000 people. The latest e-voting election was conducted on August 6, 2007 at Otama town in Fukushima prefecture. This was third e-voting election by the same town.

2.3 Typical e-Voting Operation in Japan

The following describes the typical voting procedure and types of system components used at e-voting elections in Japan.

Polling Station (During Voting Period)

(1) A voter arrives to designated polling station and hands the vote ticket to a poll worker at the reception desk.
(2) The poll worker checks the vote ticket with an electoral roll, then issues a voting card and hands the voting card to the voter.
(3) The voter moves to a voting booth, and inserts the voting card into a voting machine.
(4) The voter selects candidate on the screen.
(5) The voter touches the "cast ballot" button on the screen.
(6) The voter removes the voting card from the voting machine.
(7) The voter returns the voting card to the poll worker.

The voting card is continuously reused by reregistering for other voters.

Polling Station (After Closing the Voting Period)

(8) A poll worker removes the electronic media (CF card) from the voting machine with auditors present.
(9) The poll worker brings it to the tabulation station.

Tabulation Station

(10) The poll worker calculates voting result by accumulating data from CF cards on the tabulation machine.

After calculating voting data, an application software (e.g. M.S.Access / Excel) is used to tally up the election returns. The following diagram describes a layout of polling station (Figure 2).

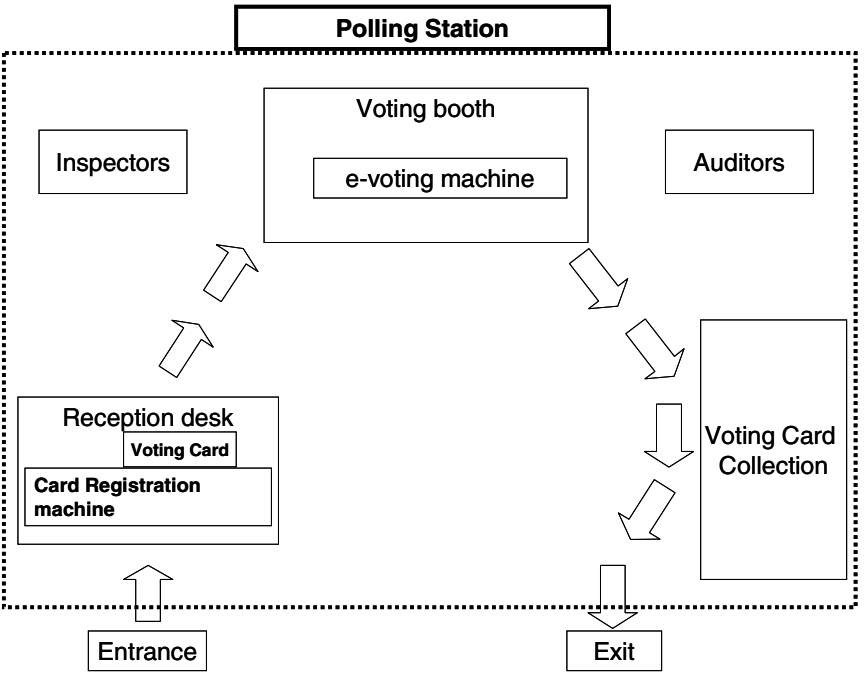

Fig. 2. Layout of a polling station and path of a voter

3 Security Analysis of e-Voting Systems

In this section, the security analysis and evaluation of the e-voting systems were conducted. First of all, we have generated our check list to be examined from past research papers and historical incident data. Then we evaluated the equipment of each manufacturer by manipulating them to go through all voting procedures. Moreover, we confirmed found security issues with the representatives from each manufacturer.

3.1 Result of e-Voting Systems Analysis

We studied seven manufacturers, which are experienced as bidders in the past elections in Japan. Then we evaluated the equipment of four manufacturers by actually operating it. The rest of three manufacturers have withdrawn from the market. Moreover, we examined one e-voting machine in detail. As a result, hardware and software of e-voting machine worked properly even with various attacks using IC card, tamper CF card data, swap the data, etc. However, it seems that parties concerned such as programmers of the

manufacturer can install malicious software in which the total result is intentionally changed.

[Examined items]
Confidentiality:

1. Confirm access control of voting programs
2. Confirm possibility of fraudulent transactions
3. Confirm traceability of voter information
4. Confirm authentication of e-voting systems
5. Confirm possibility election data disclosure

Integrity:

1. Confirm possibility of data-tampering on the CF card
2. Confirm possibility of tampering with the program
3. Confirm possibility of tampering with voting data
4. Confirm possibility of tampering with candidate information

Availability:

1. Confirm exception handling
2. Confirm error and log handling
3. Confirm back up battery
4. Confirm possibility of a Denial of Service (DoS) attack

[Result]
Confidentiality:

1. No access control was implemented at the OS level. It was possible to access the underlying OS by administration card or by connecting a keyboard to USB port embedded on the voting machine.
2. One vote results in creating one file with the same date and time. Only the names are different. The content of each file is the same for same voting candidate, so it is easy to plant files with a known candidate's string. (See Table 1).

Integrity:

1. It is impossible to open the e-voting machine while voting and to replace the data of the CF card even if we have the key of the e-voting machine. The program did not start when data remained on the CF card. Moreover, the warning lamp blinked when we pulled out the CF card, an error message is displayed, and the program stops. Therefore, it is understood that it is impossible to open the e-voting machine while voting and to replace the data of the CF card even if we have the key of the e-voting machine.
2. Since security of the tabulation machine relies on Windows Operating system, it is easy to compromise code in the e-voting machine.

Availability:

It did not react to crafted IC cards, and the e-voting machine keeps waiting for the next voter. Moreover, if the system stopped, the cause of failure is recorded on to the log file. (See Table 2). All machines are equipped with a battery as a back up power source.

Table 1. Voting data (ten trials)

File name	Size	Type	Date & Time
L1_Y55838.dat	1KB	DAT file	2007/7/3 22:00
L1_Y63803.dat	1KB	DAT file	2007/7/3 22:00
L1_Y12432.dat	1KB	DAT file	2007/7/3 22:00
L1_Y34229.dat	1KB	DAT file	2007/7/3 22:00
L1_Y59034.dat	1KB	DAT file	2007/7/3 22:00
L1_Y11221.dat	1KB	DAT file	2007/7/3 22:00
L1_Y76894.dat	1KB	DAT file	2007/7/3 22:00
L1_Y54093.dat	1KB	DAT file	2007/7/3 22:00
L1_Y99433.dat	1KB	DAT file	2007/7/3 22:00
L1_Y38965.dat	1KB	DAT file	2007/7/3 22:00

Table 2. Sample log file

Example-1
2007/06/20 16:36:08 open candidate choose screen: selected candidate
2007/06/26 16:36:12 confirmation screen: pushed confirmation button
2007/06/26 16:36:12 close screen: finish all transaction

Example-2
2007/06/06 14:00:08 open front door, close front door
2007/06/06 14:01:11 open rear door, error code: XXXXXXX

4 Security Analysis of e-Voting Operational Procedures

Operational procedures for secure e-voting is analyzed and evaluated in this section. Data for analysis was obtained through questions and interviews with the local government office.

Example of the analysis items

- The way to deliver the e-voting systems to a polling station.
- The location to keep machines prior to the election day.
- The way to test the machines.
- The way to select a manufacturer.
- Training time of the poll workers.
- Cost of election, etc.

4.1 Result of e-Voting Operational Procedures Analysis

Through the analysis we confirmed following results:

(1) The machines were kept in a depository with security guards.
(2) All machines were rented except for one.
(3) All local governments had some mechanical problems during the election.
(4) All local governments trained poll workers more than 100 hours per person.

(5) All local governments trusted the security of the manufacturer's e-voting machines without any proof of trust. They do not have procedures of software auditing for security.
(6) We found there is possibility that someone could illegally access the machines unnoticed.

Table 3. Attacker and the location of the machine

attacker \ Locati	Manufacturer Factory	→→→→ delivery	Government Warehouse	→→→→ delivery	Polling Station	→→→→ delivery	Tabulation Station
Atackers profile							
1.Manufacturer (programer etc)	O		X		X		X
2. Poll worker	X		O		O		O
3. Voter	X		X		O		X
4. Others	X		X		X		X

Those marked "O" are accessible at the given location and marked "X" are inaccessible at the given location.

5 Security Risk Analysis

We analyze the security risk of e-voting, and evaluate the vulnerabilities as follows. First, we identify assets on each machine of the e-voting systems. Second, we enumerate threats from past research paper and from past incidents confirmed by interview. Then we assess each machine and operational procedure to find vulnerabilities. Finally, we evaluate the vulnerable levels and analyze impact of risks classified as "low", "mid" or "high" risk.

5.1 Asset Identification

We identify assets from Card Registration machine, e-voting machine and tabulation machine which are components of e-voting systems as follows. (See Table 4).

Table 4. Components of e-voting systems

Machine name	Card Registration machine	e-voting machine	tabulation machine
Asset	1. Voting program 2. Voting data 3. Memory device (CF card) 4. Administrator card (IC card) 5. Candidate data 6. Election data 7. System configuration data 8. Log data 9. Operating system 10. Device driver 11. ……	1. Registration program 2. Voting card (IC card) 3. Memory device (CF card) 4. Administrator card (IC card) 5. System configuration data 6. Log data 7. Operating system 8. Device driver 9. ……	1. Tabulation program 2. Voting data of CF card 3. Memory device (CF card) 4. Candidate data 5. Election data 6. System configuration data 7. Log data 8. Operating system 9. Device driver 10. ……

5.2 Threat Clarification

We enumerated 68 threats from past research papers, incidents and interviews. The threats are categorized as software, hardware and operation. (See Table 5).

Software threats: 28 (e.g. to replace voting data, log data)
Hardware threats: 26 (e.g. to insert crafted voting card, denial of service)
Operational threats: 14 (e.g. operational mistake, authentication bypass)

Table 5. Example of threat models

Machine	Operational procedure
USB port of machine is accessible to anyone. One voting card allows multiple votes. Power outage. Unexpected Interruption of Operating System. Bugs on a voting program.	Malicious program insertion by poll worker. Installation of backdoor program on machines. Authentication bypass by password Tampering of voting data with administration card.

5.3 Vulnerability Assessment

Through the analysis above, we identified vulnerabilities which result in insecure e-voting. (See Table 6).

Table 6. Vulnerability of systems and operational procedure

Vulnerability of machine	Vulnerability of operational procedures
A. Ports (USB, LAN) are accessible. B. Power backup (battery) is not equipped. C. No physical protection of hardware. D. No access control at operating system. E. Others	A. Machines are not under guard in the polling station. B. Operation manuals are not prepared. C. Poll workers are not trained. D. Power backup (generator) is not equipped. E. Others.

5.4 Risk Level Determination

We analyze an impact of risks from the following concepts.

Risk = **Asset** Φ **Threat** Φ **Vulnerability**

We evaluate the risk based on the following criterion.

high: both machine and operational procedures are vulnerable.
mid: either machine or operational procedures is vulnerable.
low: neither machine nor operational procedures is vulnerable.

In conclusion, we evaluated the security risk analysis as follows. (See Table 7).

Table 7. Risk analysis results

[e-voting machine] C: Confidentiality, I: Integrity, A: Availability

Asset (detailed explanation)	C I A	Threat	Vulnerability: Machine	Vulnerability: Operational procedures	Risk
Voting program (software running on the machine)	C	Logical error(bug) in the program	---	**---**	**high**
	C	Insert of malicious code in the program	---	---	**high**
	I	Tampering by programmer or poll worker	---	---	**high**
	A	Power interruption	B	D	low
Voting data of CF card (designated data by voters)	C	Spoofing, theft	C	A	low
	I	Tampering by programmer	C	A	low
	A	Hardware trouble	E	B	low
Administrator card (IC card) (access primary mode, OS mode)	C	Lost, stolen, Privilege	E	A	low
	A	Physical break, read error	E	E	low

[Card Registration machine]

Asset (detailed explanation)	C I A	Threat	Vulnerability: Machine	Vulnerability: Operational procedures	Risk
Registration program (software running on the PC)	C	Logical error(bug) in the program	---	---	**high**
	C	Insert of malicious code in the program	---	---	**high**
	I	Tampering by programmer or poll worker	---	---	**high**
	A	Power interruption	B	D	low
Voting card (IC card) (authentication card to vote)	C	Spoofing, theft	C	A	low
	I	Tampering by programmer	C	A	low
	A	Hardware trouble	E	B	low

[Tabulation machine]

Asset (detailed explanation)	C I A	Threat	Vulnerability: Machine	Vulnerability: Operational procedures	Risk
Tabulation program (software running on the machine)	C	Bug(error) in the program	nothing	nothing	**high**
	C	Embedded code in the program	nothing	nothing	**high**
	I	Tampering by programmer or poll worker	nothing	nothing	**high**
	A	Power cut	B	D	low
Voting data of CF card (designated data by voters)	C	Spoofing, stolen	C	A	low
	I	Tampering by poll worker	C	A	low
	A	Hardware trouble	E	B	low
	A	Physical break, read error	E	E	low

6 Conclusion

In this research, we clarified the threat of e-voting systems and also, made a check list and examined e-voting systems based on known vulnerabilities. Then, we examined the operational procedure of e-voting and evaluated risk level. Finally, we found out possible vulnerabilities of e-voting systems. The following are the main risks found in our research. They are related to each machine program prepared by the manufacturers and poll workers.

(1) All programs have not been audited.
    Logical errors or malicious programs may lead to problems.
(2) Security of e-voting systems relies on the operating procedures.
    A malicious poll worker could install malware.
    The tabulation program could be tampered by a poll worker.
(3) Other Risks
    No verification scheme for confirming voting data.

The above risks of e-voting have not occurred yet in reality. However, in the future, the more we use e-voting, the more possibilities that problems will arise. Since the weakest link in the chain of e-voting today is the secure operational procedures, we suggest that the e-voting system itself improve and add strict restriction so that even with poor operation it will be secure.

Suggestions for Improvement

(1) When e-voting systems, such as e-voting machines, card Registration machines and the tabulation machine are shipped to the polling station, the equipment should be checked by a third party.
(2) All open ports (USB, LAN, etc.) that the e-voting machine has should be sealed after the program is installed.
(3) Digital signatures for the tabulation program and the card registration program should be required.

References

[1] The e-voting special law: Japan Government committee of vote (2002)
[2] Kohno, T., Stubblefield, A., Rubin, A.D., Wallach, D.S.: Analysis of an Electronic Voting System in IEEE Symposium on Security and Privacy 2004. IEEE Computer Society Press, Los Alamitos (2004)
[3] Feldman, A.J., AlexHalderman, J., Felten, E.W.: Security Analysis of the Diebold Accu Vote-TS Voting machine
[4] Rebecca Mercuri's Statement on Electric Voting (2001 and 2011), http://www.notablesoftware.com
[5] Rubin, A.: Security Considerations for Remote Electronic Voting over the Internet AT&T Labs-Research
[6] California secretary of State Bill Jones: A Report of the Feasibility of Internet Voting

[7] Smith, R.G.: Electronic Voting: Benefits and Risks Australian Institute of Criminology (2002)
[8] Jones, D.W.: The case of the Diebold FTP site Part of the Voting and Elections web pages by Dougls W. Jones (2003), http://www.cs.uiowa.edu/ jones/voting/dieboldftp.html
[9] Barry, C., Dacey, P., Pickering, T., Byme, D.: Electric Voting and Electronic Counting of Votes (2006)
[10] Willams, B. J.: Security in the Georgia Voting System
[11] Shiraishi, A., Tanaka, T.: Polling-site electronic voting system. In: 19th technical conference (2000)
[12] Verified Voting Foundation: Voter's Guide to Electronic Voting
[13] Manhattan Borough President C. Virginia Fields and The Center for Independence of the Disabled in New York,Inc: Voting Technology for people With Disabilities
[14] Open Rights Group: 2007 Election Report
[15] Yee, K.-P., Wagner, D., Hearst, M., Bellovin, S.M.: Prerendered User Interfaces for Higher-Assurance Electronic Voting,
http://www.usenix.org/events/evt06/tech/full_papers/yee/yee_html
[16] Election Data Services, Inc: Election Day Survey Report, Part 2 Survey Results (2004)
[17] Iwasakai, M.: e-voting: Nippon keizai sha (2004)

Bingo Voting: Secure and Coercion-Free Voting Using a Trusted Random Number Generator

Jens-Matthias Bohli[1,⋆], Jörn Müller-Quade[2], and Stefan Röhrich[2]

[1] NEC Laboratories Europe, Network Research Division,
Kurfürsten-Anlage 36, 69115 Heidelberg, Germany
bohli@nw.neclab.eu
[2] Institut für Algorithmen und Kognitive Systeme / E.I.S.S.,
Universität Karlsruhe (TH), 76128 Karlsruhe, Germany
{muellerq,sr}@ira.uka.de

Abstract. It is debatable if current direct-recording electronic voting machines can sufficiently be trusted for a use in elections. Reports about malfunctions and possible ways of manipulation abound. Voting schemes have to fulfill seemingly contradictory requirements: On one hand the election process should be verifiable to prevent electoral fraud and on the other hand each vote should be deniable to avoid coercion and vote buying.

This work presents a new verifiable and coercion-free voting scheme *Bingo Voting*, which is based on a trusted random number generator. As a motivation for the new scheme two coercion/vote buying attacks on voting schemes are presented which show that it can be dangerous to let the voter contribute randomness to the voting scheme.

A proof-of-concept implementation of the scheme shows the practicality of the scheme: all costly computations can be moved to a non time critical pre-voting phase.

Keywords: Secure electronic voting, coercion-free, receipt-free.

1 Introduction

Elections have to meet a lot of requirements, e.g., the German constitution speaks about the selection of the members of German House of Representatives in general, direct, free, equal, and secret elections[1]. For security considerations of voting protocols, mainly the last three properties are of interest: An election should be free, i.e., nobody can be coerced to cast a certain vote, it should be equal, i.e., nobody can influence the result more than with her own vote, and it should be secret: no one is able to learn the votes of other people.

Traditional voting schemes using paper and ballot boxes cannot be trusted to guarantee all these security properties. Ballot stuffing, miscounting, and the manipulation or destruction of votes during tallying are possible. Current voting

⋆ Work done while the author was at Universität Karlsruhe (TH).

[1] Grundgesetz Art. 38(1): "Die Abgeordneten des Deutschen Bundestages werden in allgemeiner, unmittelbarer, freier, gleicher und geheimer Wahl gewählt."

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 111–124, 2007.

machines cannot be considered to be a secure solution as studies about machines used in practice [1,2] showed.

These problems led to an increasing interest in voting schemes which allow the voter to verify that her vote was counted. However, such a proof should be meaningful only for the direct recipient, because otherwise coercion and vote buying become substantially simplified. Such schemes are called coercion-free or receipt-free[2].

An additional important requirement for voting schemes is usability. A scheme must be convincing in a very direct way and one cannot expect all voters to use electronic devices apart from the voting machine. This makes the design of a voting scheme even more difficult, because many cryptographic techniques cannot be used to directly convince humans.

Our Contribution

In this work we propose a new voting scheme, called Bingo Voting due to the use of a random number generator, comparable to a bingo cage. The new scheme achieves:

- Ballot casting assurance and universal verifiability, i.e., the voter can check if her own vote is cast and counted as intended, and everyone is able to verify that all votes are correctly counted as recorded on a bulletin board without learning the content.
- Depending on the binding property of the commitments used the scheme offers either everlasting privacy or unconditional correctness.
- Coercion-freeness, i.e., even if the voter deviates from the protocol she does not gain any evidence which allows her to prove anything about the contents of her vote.

Security properties like anonymity or eligibility (i.e., one vote per eligible voter) are, in contrast to purely electronic voting schemes, easily obtained by traditional methods. The authorization is handled in front of the voting booth and an eligible voter may enter once to cast his vote. The voting machine reorders the votes and has to be trusted in order to guarantee anonymity.

The voting scheme offers a very high usability. Only very limited capabilities on the side of the voter are required. The voting process corresponds to the voting with today's voting machines: the voter has to press the button that is assigned to the intended candidate. To ensure the correctness of her vote, the voter only needs to check equality of two random numbers and check if her paper receipt has been posted to a bulletin board. The scheme remains secure if not all voters actually verify the process as long as the attacker cannot predict which voter actually will be verifying.

The security properties listed above are achieved relative to very realistic assumptions:

- A non interactive commitment scheme with some homomorphic properties is needed, e.g., Pedersen commitments [3]. If general zero-knowledge protocols

[2] The term receipt-free might be misleading as the voter indeed obtains a receipt.

are used in the post-voting phase, then these homomorphic properties are not needed. Furthermore, if one is willing to use check samples instead of giving proofs then even physical commitments, e.g., using strong boxes, become possible.[3]
- A trusted random number generator is needed.
- If a voter should not only be able to detect cheating attempts, but also to prove an electoral fraud, then the printed receipts should be difficult to forge.

To show the practical applicability of the new scheme it has been implemented in Java as a proof-of-concept.

Furthermore we present two new coercion/vote-buying attacks on existing schemes [4,5,6,7] which strongly suggest that the voter should not be trusted to contribute her own randomness. This gives an additional motivation for the use of trusted random number generators.

2 Related Work

Several voting schemes have been proposed over the last years.

Protocols that require the user to provide random choices additional to the actual vote [4,5,6] will be analyzed in the following section. These protocols allow a practical attack which is overlooked in cryptographic models as it exploits the limited memory of humans involved in the voting process. The Mark-Pledge scheme of [8] does not associate the human generated randomness to a specific candidate and is not vulnerable to these attacks. However, voters have to commit to random choices which introduces new assumptions or makes the scheme impractical. ThreeBallot voting [9] also demands additional random choices from the voter and, as already outlined in the paper, a coercion attack becomes possible.

In Section 2.2 we will take a closer look at Punchscan [7,10] and point out an attack if the choice of a layer (step E.1 in [11]) is not done before the voter receives her ballot (as it was the case in earlier versions of Punchscan [10]). This shows possible ways of coercion, when the voter has to make decisions after getting unpredictable input. Examples for further protocols having the usage of cryptographic paper ballots in common are Prêt à Voter [12], Scratch & Vote [13], and Benaloh's simple verifiable elections [14].

2.1 Voting Schemes Using the Order of User Inputs

Neff [4] proposed a voting scheme which is based on the temporal order of the interaction between the voting machine and the user. Other schemes use the same idea, e.g., the scheme by Reynolds [5]. Moran and Naor present a scheme which ensures everlasting privacy [6] and they prove the coercion-resistance of their scheme in a simulation-based model.

[3] However, samples instead of proofs would weaken the coercion-freeness as a small fraction of votes is opened.

The basic concept behind these schemes is, that the voting machine commits itself to some random values, e.g., by printing it on a receipt without showing it to the voter. After the machine is committed, the voter casts her vote and enters some randomness into the machine. This randomness is then used to generate a proof that the vote will be counted for the chosen candidate. In order to avoid coercion, the voter can also input random values for the other candidates, but this is done *before* the voting machine is committed, so the voting machine is able to produce fake proofs for these candidates. On a receipt every candidate is printed with the corresponding user choices. Crucial for these schemes is the order of the interaction, first the user enters dummy values for the other candidates, then the voting machine commits, after this the voter enters the random value for the real candidate, so that the machine cannot fake the proof for the real vote.

A "Babble" Attack. Because of the limited memory of the human brain there may be the possibility of a vote-buying/coercion attack, if the scheme doesn't restrict the length of the random choices by the voter to very short lengths or there is a large number of candidates. Suppose the attacker provides the voter with an ear piece, i.e., a small radio receiver which allows the voter to listen to the attacker in the voting booth. The adversary now remotely babbles a long stream of random choices through the ear piece to the voter. The attacker coerces the voter to vote for a special candidate and to enter the random values she hears through the ear piece in the correct chronological order. Because the random choices for the real vote are entered last (and this is required by the schemes) the attacker can check if the random choice for the chosen candidate really was sent after the choices used for the other candidates.

This attack requires that there are sufficiently many random choices such that the voter can't memorize and rearrange the random values. Even if the voter had access to a pen and a paper in the booth the attack remains dangerous as the attacker could observe the time the voter spends in the voting booth: For very many or very long random values and an appropriate timing when this information is sent over the ear piece it takes time to write down all the random choices before starting to vote.

This attack might seem unrealistic and only a special case of using a video camera to document the complete voting procedure of a coerced voter. But because an ear piece can be very small and requires only one-way communication (or no communication, if devices like a sealed audio-player with no user control are used) we think this attack shouldn't be ignored. Especially, the babble attack points out, that coercion-free voting schemes should clearly state the amount of isolation which is assumed in the voting booth, if there must be more protection than preventing the attacker to see what happens there.

2.2 Punchscan

Paper Ballot Scheme. Punchscan [7,10,11] is a paper based voting scheme that recently attracted attention. A Punchscan ballot consists of two paper sheets

that are attached one upon the other (see Figure 1 for sample ballots). On the top page, the list of candidates is given, assigned to each candidate is a letter in a random permutation. The upper sheet has several holes through which letters from the lower sheet (in randomized order) are visible. To vote, the voter will mark the letter assigned to her candidate such that the mark is visible on both sheets. The voter can choose one layer of her ballot as a receipt to take home. A system of published commitments and reveals allows the voter to verify the tally without being able to prove what she voted for. For more details, we refer to the Punchscan webpage [7].

A Vote Buying Attack. One way of vote buying is possible even though the receipts do not reveal the actual vote. A vote buyer may offer a reward for a certain top and bottom layer, respectively. Considering a contest with two choices, say, YES or NO, a vote buyer interested in convincing people to vote for NO might offer to pay for

- A top receipt where YES is assigned to the letter A and the left bullet is marked, or
- A bottom receipt where A appears in the left bullet and is marked.

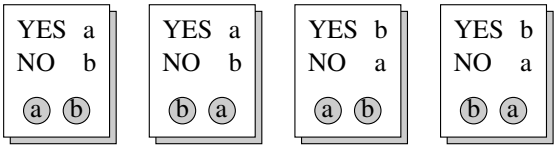

Fig. 1. Possible Punchscan ballots for a two-candidate race

Both layers together constitute a YES-vote. A voter who votes YES might hold both layers together, a top and bottom layer that would entitle her for the reward. However, she has to shred one of her papers. A voter for NO will have the benefiting layers on separate ballots, thus, being able to choose the corresponding receipt. To maximize the probability of the payment, voters are motivated to vote with NO. By enumerating all possible ballots, see Figure 1, this becomes obvious: only in the first possible ballot the voter can qualify for the reward by voting YES. In the next two of the four possible ballots, the voter can qualify for the reward by voting NO, and for the last ballot, it is impossible to produce one of the rewarding layers. Voters following the attack unreservedly will vote in favour of the coercer with probability of $1/2$, vote against the coercer with probability $1/4$ and vote according to their own decision with probability $1/4$.

The success probability and the effect of this attack diminishes when more choices are available. Even though this attack is not published there is evidence that this attack is known to the designers of Punchscan, because a change was made which prevents this attack in [11] in contrast to [10]: In the first step of the election phase the voter is required to state her choice (top receipt or bottom

receipt) before she can see the ballot. We think the above attack might be the motivation behind this step, as a connection to chain voting as indicated in [11] does not seem obvious.

This attack may seem similar to a vote randomization attack in which a voter should always mark the left bullet and which is still possible in the modified version. But the attack described here increases the probability to vote for a specific choice and therefore is much stronger.

3 Bingo Voting

3.1 Basic Idea

A basic idea for a verifiable vote is to have ballots with a unique serial number. Each voter will cast her vote on the ballot and take a copy of the ballot home. When the election result is published, all ballots are published such that every voter can check that her vote is indeed counted for the intended candidate. Unfortunately, this simple voting protocol paves the way for vote buying or coercion. The voter can easily prove her vote by showing her ballot.

However, the voter were able to deny her vote if she could fabricate and present a fake ballot that is published in the list of all votes. Such a fake ballot can be imagined as the vote of another voter or a *dummy vote*—a vote that appears in the list of votes but does not count.

To avoid problems that arise from the approach of swapping receipts with other voters our scheme makes enough dummy votes available to give to every voter a receipt for every (not elected) candidate in addition to the receipt for the elected candidate. All the serial numbers will be printed on the same receipt, each number printed next to the corresponding candidate in one line (see the Vote Receipt in Figure 2 for an example). Thus, a vote is simply represented by a number.

The tallying of the real votes taking into account the dummy votes will be assured by means of cryptography. To achieve this the dummy votes will be chosen out of a pool of random numbers that are committed before the election starts. The voter must be assured, that her actual vote—the number of the real ballot—is not a dummy vote. This is the case if the voter can actually witness the random generation of the fresh number that will represent her vote, while the dummy votes were previously determined and committed. In the polling booth, the voter has only to check, that the fresh number appears on the receipt as intended. The cryptographic proof that every ballot contains only one real vote and that dummy votes do not count is published with the tally and can be checked afterwards.

In the following, we describe the protocol in a more formal and detailed way.

3.2 Preliminaries

The scenario we assume in the following is a poll with ℓ candidates and n eligible voters. To ease the description of the scheme we restrict to the case of a single

voting machine, but an extension to multiple machines is straightforward. In the polling booth is a trusted random number generator (RNG) and the voting machine available.

3.3 Pre-voting Phase

Before election day, the voting machine will generate for every candidate P_i n random numbers N_j^i, yielding dummy vote pairs (N_j^i, P_i) for the candidate. Together, $m = n \cdot \ell$ dummy votes are created for the candidates $P_1, \ldots, P_\ell$. Unconditionally hiding commitments[4] $C_1, \ldots, C_m$ to the dummy vote pairs are computed using random coins r_j^i. The commitments $C_j^i = \mathsf{commit}((N_j^i, P_i); r_j^i)$ are shuffled and published on a bulletin board before the election starts. Additionally, the equal distribution of the committed dummy votes to the candidates is proven without opening the commitments[5].

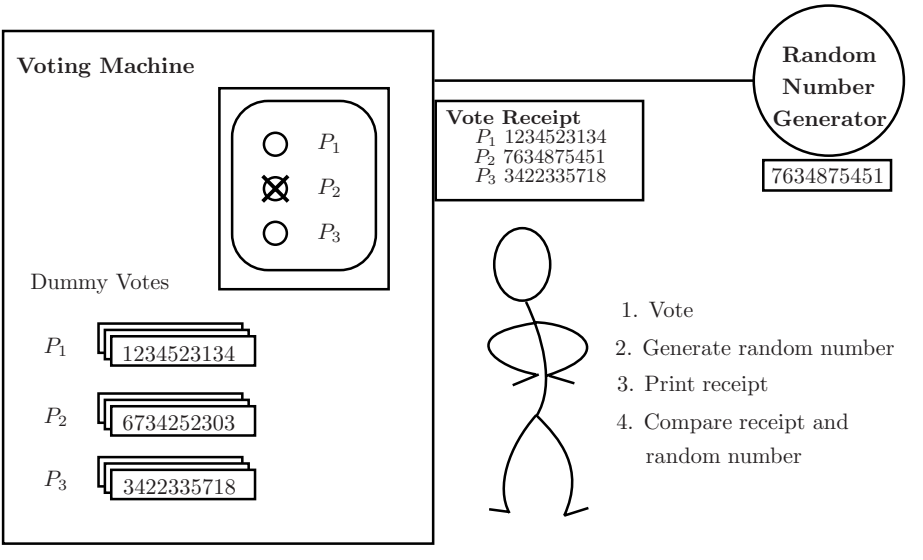

Fig. 2. Voting phase

3.4 Voting Phase

In the voting booth, a voter has to perform the following steps to cast a vote, which are also shown in Figure 2:

[4] Unconditionally hiding commitments yield everlasting privacy. To achieve unconditional correctness unconditionally binding commitments would have to been used.

[5] This can be done via randomized partial checking [15] if the part of the commitment containing the candidate can be opened without revealing the random number.

1. The voter will first indicate her vote by pressing the according candidate's button on the voting machine.
2. Next the random number generator generates a fresh random number R.
3. The random number R is transferred to the voting machine and assigned to the candidate of the voter's choice.

   For each other candidate, the voting machine will draw randomly one number out of the pool of dummy votes for the respective candidate.

   The machine will print out a receipt listing candidates and numbers: The candidate that was voted is assigned the fresh random number R, for the other candidates the respectively chosen dummy vote is shown. Figure 2 contains a sample receipt.
4. The voter has to verify that the number shown on the random number generator is assigned to the party she intended to vote for. If this is not the case, the voter has to protest immediately.[6]
5. The voter leaves the booth and takes out the receipt. For any outsider it is impossible to recognize the fresh random number and therefore the vote this ballot implies.

3.5 Post-voting Phase

After the election the voting machine calculates the result and sends it together with a proof of correctness to a public bulletin board. The published data consists of four sections:

1. The final outcome of the poll;
2. A lexicographically sorted list of all receipts issued;
3. A list of all unused dummy vote/candidate pairs with the respective commitment and reveal information;
4. Non-interactive zero-knowledge proofs[7] for the correctness, i.e., that the dummy vote of every unopened commitment was indeed used on one receipt.

Now every voter can easily verify that her receipt is included in the list and therefore was counted for the tally. Every voter can verify that the number of remaining commitments is as expected (i.e., every vote for a candidate causes a fresh random number, so one dummy vote isn't needed. Therefore, if a candidate has a votes and b voters were absent, the candidate should have $n-a-b$ remaining commitments.) and the reveal information leads to the given tally. Finally, the non-interactive proof for the correctness of the receipts can be checked.

In the following we describe a proof for the correctness of the receipts which is efficient and uses standard shuffling techniques[8] which are common in e-voting

[6] See also the discussion in Section 4.

[7] For efficiency reasons we will use "proofs" with 50 % soundness in this paper as many votes need to be manipulated to change the outcome of an election. To actually have zero-knowledge proofs one would need several iterations.

[8] Commitments are shuffled and later revealed [16]. To prove the correctness of the shuffle we apply the method of *randomized partial checking* [15] to every single ballot.

protocols. For this, we need commitments with a special homomorphic property: it should be possible to generate a new "masked" or re-randomized version of a commitment, which is a new commitment to the old value (using new randomness), and the correctness of this masking should be provable. Using Pedersen commitments [3] it is easy to generate such masked commitments and to prove the equality of the old and new values.

The proof shows that all unopened (used) commitments are assigned to receipts. By a counting argument, the revealed unused commitments correspond to the election result (and a constant offset for absent voters). It has the following steps for every single receipt:

- A new commitment C to (R, P) is generated, where P is the really chosen candidate and R the output of the trusted RNG.
- The $\ell-1$ commitments to dummy values used for this receipt and C are published (without revealing). This list of commitments is called $\mathcal{C}_{left}$, everyone can check that $\ell - 1$ previous published commitments are used.
- The commitments are masked to new commitments to the same value and shuffled. The new list $\mathcal{C}_{middle}$ is published.
- The last step is repeated with the commitments of $\mathcal{C}_{middle}$, yielding a new list $\mathcal{C}_{right}$ which is published.
- The commitments of $\mathcal{C}_{right}$ are revealed. The contents must equal the corresponding numbers on the receipt.
- A random bit s is chosen[9]. If the bit s is 0, the association between the commitments of $\mathcal{C}_{left}$ and $\mathcal{C}_{middle}$ is published and the correctness of the masking is proven without revealing the commitments of these lists[10]. If s equals 1 the according relation between commitments of $\mathcal{C}_{middle}$ and $\mathcal{C}_{right}$ is proven.

Any manipulation in the proof for a single receipt affects at most ℓ votes and is detected with a probability of 50 % per manipulated receipt. If a higher assurance is necessary (e.g., if the vote counts of the candidates are very close to each other), the proof for the single receipts can be repeated to yield higher soundness.

Our scheme scales rather well, after the creation of $n\ell$ dummy commitments in the pre-voting phase, additionally q commitments are created for the real votes. During the proof, $q\ell$ commitments have to be masked twice to yield new commitments to the same value.

[9] The randomness and unpredictability for the voting authority is important for the correctness property. It can be achieved by using a trusted random number generator in public or by other means, some of them are discussed in [6]. The most practical method probably would be using the Fiat-Shamir heuristic to handle the secure hash of the whole transcript to that point as randomness, but solutions like using physical sources or a coin flipping protocol are also possible.

[10] For Pedersen Commitments this is simply done by publishing the random choices used for the re-randomization.

4 Attacks, Assumptions and Security

Different assumptions are necessary: The main security goal is certainly to ensure *correctness* of the election result. The key assumption to ensure correctness is a trusted random number generator attached to the voting machine. It should provide some clearly evident form of randomness, we even recommend a mechanical device, similar to devices used in lotteries or to a bingo cage, and read the random number by sensors.

In order to provide *coercion protection*, we must rely on the voting machine: The voting machine must not be tampered with and must guarantee the secrecy of votes. Likewise the voting booth has to be secured, e.g., no hidden cameras may be able to surveil the voting (this would already be a threat for classical elections).

Table 1. Assumptions of Bingo Voting

Correctness	RNG is uncorrupted
Coercion protection	RNG is uncorrupted **and** (voting machine is uncorrupted **or** (voting machine is only passively corrupted **and** actions on the voting machine/receipts are not linked to voters))

Before we discuss the security in more details, Table 1 summarizes the assumptions.

4.1 Correctness

At the start of the election it is proven that each candidate has the same number of dummy votes. For each vote a fresh random number is generated and associated with the candidate voted for. Hence for each vote for a specific candidate one dummy vote for this candidate is left unused. A corrupted voting machine could generate a fresh number to a candidate of its choice if the fresh random number equals a dummy vote, but the probability for this equality is negligible for a trusted random number generator[11]. The fresh number cannot be associated to another candidate, because this is checked by the voter in the booth. Also no additional fresh numbers can be introduced as the zero-knowledge proof guarantees $\ell - 1$ dummy votes on each ballot.

Fraud should not only be detected, but there should be evidence of the fraud. Assume the voting machine will cast a vote for a candidate that was not intended by the voter, however, pretends to behave correctly. Our protocol makes this evident to the voter immediately in the polling booth: the number that is shown on the RNG is not printed on the right place. In this case the voter can

[11] For the correctness the random numbers need not be uniformly distributed. It is sufficient if the attacker cannot predict the numbers.

immediately be given a possibility to cancel the previous vote and revote (This will, however, need suitably more commitments to be prepared in the pre-voting phase. Additional care has to be taken to avoid denial of service attacks by an always complaining voter.). Each voter can check if her receipt appears on the list of ballots which is published after the election. If a voter does not find her receipt in the list, she must be able to complain. We intend therefore a trusted printer, which uses unforgeable paper to allow the voter to prove that she indeed has a valid receipt.

According to the definitions in [8] we achieve *cast as intended* by comparing and checking the receipt and the output of the random number generator by the voter, the vote is *recorded as cast* can easily be checked through comparison with the public list of receipts, and the property *counted as recorded* is achieved by the (public) proof at the end of the election, so that *universal verifiability* is reached. Through the *end-to-end* and *direct verification* of "cast as intended" and "recorded as cast" we achieve a non-immediate notion of *ballot casting assurance*. For a really direct and immediate verification the public list of receipts has to be updated with new votes and their proofs in real time. This is possible (probably with the addition of some helper organisations like the proposal for MarkPledge in [8]), but it would ease vote coercion attacks by an attacker with access to a corrupted voting machine.

4.2 Coercion Protection

The vote receipt cannot be used to prove anything about the contents of a vote to a third party. The hiding property of the commitments in the published dummy votes together with the information of the zero-knowledge proof does not allow an adversary to distinguish the random number of the trusted random number generator from the dummy votes which were stored in the voting machine and published as a commitment. Therefore, the random number generator has to be trusted for coercion protection, too, so that it doesn't generate numbers which can be recognized by an attacker.

In order to gain privacy and coercion protection you also have to trust the voting machine. It mustn't be actively corrupted, e.g., the addition of a secret camera together with the recording of the votes would clearly violate privacy. As with most voting machines, coercion attacks are also possible if you only change the software of the machine, like adding a secret command to the software of the voting machine, which is activated by some special keystrokes which a voter is coerced to press, and which votes in favour of the attacker and records this.

However, another relevant attack can to a certain extend be prevented, even if the adversary has access to the secret commitment information or passively corrupted the voting machine (i.e., he learns every information the voting machine knows, but doesn't change the voting machine or its software). The secrecy of a vote can be maintained as long as the attacker cannot obtain the specific receipt of the voter or otherwise link the voting process to an individual voter. To protect voters from this some non-government organisations (NGOs) could collect paper receipts to check on the correctness of the vote. If it is possible to

anonymously dispose the paper receipt the voter cannot be coerced afterwards and the correctness of the vote will still be checked with a high probability unless many of the NGOs are corrupted.

The concrete voting machine has to be designed very carefully in order to avoid attacks comparable to side-channel attacks, e.g., the state of the voting machine (like if it's waiting for input or printing a receipt) should be hidden from persons outside of the voting booth and the output of the random number generator and the presentation of the receipt to the voter should be very close in time. This avoids a kind of "reverse" babble attack where the voter is forced to use a microphone and read the numbers presented to him.

5 Implementation

We implemented a proof-of-concept implementation in Java [17]. Pedersen commitments [3] were used as described for the underlying commitment scheme, therefore everlasting security is achieved. Our straightforward implementation in Java, which is not optimized for speed, needs about 0.6 seconds per potential voter (assuming a voter participation of 80 % and five candidates) for all three phases together on a standard 3 GHz Pentium 4 CPU and a bit length of 1000 for the commitments. Most of the time consuming work, generation of and calculations with the commitments, can be separated to the pre-voting phase, where speed is not a very important factor. Also the output size is acceptable, even including some debugging information, repeating many things and writing the commitments as ASCII encoded decimal numbers under 10 KBytes per potential voter are used. This enables a voter to download the whole information needed to check the proof of her electoral district.

The prototype is mainly a demonstration for the feasibility of computation time and size of the proofs, a more detailed system, especially with a real hardware number generator for the voting booth, still has to be realized in order to analyze the whole voting process from a systems perspective as done for other schemes in [18,19].

6 Outlook

Further research is also required to enhance the usability and reduce the administrative requirements of voting schemes. Possibly, a combination of existing schemes, like the scheme of Moran and Naor [6] and our Bingo Voting, would be an improvement in this respect.

The combined scheme would have a user interface similar to our Bingo Voting scheme. The voter's only input is the choice of a candidate. After the voter has chosen her candidate, a random number is generated by a trusted random number generator and transferred to the voting machine. The machine prints a receipt containing the name of each candidate with an associated random number generated by the voting machine itself for the not elected candidates and the number of the random number generator for the elected candidate. The voter

has to check if the number of her candidate corresponds to the output of the random number generator. The voting machine commits to proof information before the generation of the trusted random number by printing it on the receipt and the voter has to check that it is done before the random number generator is started. This information is used to generate proofs of the election result as in [6].

This new scheme avoids the babble attack and even requires a slightly weaker assumption for the printer compared to [6], because the commitment does not have to be hidden from the voter. Still, it must be assumed, that the voting machine cannot exchange or generate the commitment after receiving the trusted random number. Compared to Bingo Voting, this assumption avoids the administrative effort of generating and storing the dummy votes. However, the voter has to be more careful to detect *message reordering attacks*[12] [18] which may affect the correctness property, while Bingo Voting isn't vulnerable to this kind of attacks to forge elections (but a coercion attack might be possible through this corruption of the voting machine).

7 Conclusions

We have shown, that many voting protocols where the user needs to make decisions beyond choosing one candidate are susceptible to coercion attacks. We could reveal new coercion attacks to recently proposed voting protocols. To avoid those attacks, we introduce the assumption of a trusted random number generator inside the polling booth. We have presented a protocol basing on a random number generator, that makes it easy for the voter to vote and check correctness of the vote and have demonstrated that implementing our scheme is practical.

An open problem remaining is to find a suitable and realistic model for the treatment of voting protocols. Attacks like the babble attack are not covered by any security model known to us, even enhancements of very strong simulation based models to handle coercion ignore such a threat.

References

1. University of California: Reports of top-to-bottom review of voting machines (2007), http://www.sos.ca.gov/elections/elections_vsr.htm
2. Gonggrijp, R., Hengeveld, W.J., Bogk, A., Engling, D., Mehnert, H., Rieger, F., Scheffers, P., Wels, B.: Nedap/Groenendaal ES3 voting computer – a security analysis (2006), http://www.wijvertrouwenstemcomputersniet.nl/images/9/91/Es3b-en.pdf
3. Pedersen, T.P.: Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)

[12] In such an attack a corrupted voting machine maliciously executes the protocol steps with the human in the wrong order to be able to fake a vote.

4. Neff, C.A.: Practical high certainty intent verification for encrypted votes (2004), http://www.votehere.net/vhti/documentation/vsv-2.0.3638.pdf
5. Reynolds, D.J.: A method for electronic voting with Coercion-free receipt. In: FEE 2005 (2005), http://www.win.tue.nl/~berry/fee2005/presentations/reynolds.ppt
6. Moran, T., Naor, M.: Receipt-Free Universally-Verifiable Voting With Everlasting Privacy. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 373–392. Springer, Heidelberg (2006)
7. Chaum, D.: Punchscan (2006), http://punchscan.org/
8. Adida, B., Neff, C.A.: Ballot Casting Assurance. In: EVT 2006. Proceedings of the First Usenix/ACCURATE Electronic Voting Technology Workshop, Vancouver, BC, Canada (August 1, 2006), http://www.usenix.org/events/evt06/tech/full_papers/adida/adida.pdf
9. Rivest, R.L.: The ThreeBallot Voting System (2006), http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf
10. Popoveniuc, S., Hosp, B.: An Introduction to Punchscan. In: VSRW 2006. Threat Analyses for Voting System Categories, A Workshop on Rating Voting Methods (2006), http://vote.cs.gwu.edu/vsrw2006/papers/9.pdf
11. Popoveniuc, S., Hosp, B.: An Introduction to Punchscan. In: IAVoSS Workshop On Trustworthy Elections, WOTE 2006 (2006), http://punchscan.org/papers/popoveniuc_hosp_punchscan_introduction.pdf
12. Chaum, D., Ryan, P.Y., Schneider, S.: A Practical Voter-Verifiable Election Scheme. In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 118–139. Springer, Heidelberg (2005)
13. Adida, B., Rivest, R.L.: Scratch & Vote: Self-Contained Paper-Based Cryptographic Voting. In: WPES 2006. Proceedings of the 5th ACM workshop on Privacy in electronic society, pp. 29–40. ACM Press, New York (2006)
14. Benaloh, J.: Simple Verifiable Elections. In: EVT 2006. Proceedings of the First Usenix/ACCURATE Electronic Voting Technology Workshop, Vancouver, BC, Canada (August 1, 2006), http://www.usenix.org/events/evt06/tech/full_papers/benaloh/benaloh.pdf
15. Jakobsson, M., Juels, A., Rivest, R.L.: Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking. In: USENIX Security Symposium, pp. 339–353 (2002)
16. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM 24, 84–88 (1981)
17. Sun Microsystems: Java Platform, Standard Edition (2006), http://java.sun.com/
18. Karlof, C., Sastry, N., Wagner, D.: Cryptographic Voting Protocols: A Systems Perspective. In: Proceedings of the Fourteenth USENIX Security Symposium (USENIX Security 2005), pp. 33–50 (2005)
19. Ryan, P.Y.A., Peacock, T.: Prêt à Voter: a Systems Perspective. Technical Report CS-TR-929, School of Computing Science, University of Newcastle (2005), http://www.cs.ncl.ac.uk/research/pubs/trs/papers/929.pdf

Enhancing the Trust and Perceived Security in e-Cognocracy

Joan Josep Piles[1], José Luis Salazar[1], José Ruíz[1], and José María Moreno-Jiménez[2]

[1] Dpto. de Ingeniería Electrónica y de Comunicaciones, Universidad de Zaragoza
jpiles@unizar.es, jsalazar@unizar.es, jruiz@unizar.es
[2] Grupo Decisión Multicriterio Zaragoza, Universidad de Zaragoza
moreno@unizar.es

Abstract. e-Cognocracy is a new, creative, innovative and cognitive democratic system based on the evolution of living systems which focuses on the extraction and social diffusion of the knowledge derived from the scientific resolution of highly complex problems associated with public decision making related to the governance of society. Among the many tools needed to fully develop e-cognocracy, we will focus in e-voting, as it is the first needed to gather the information supplied by the citizens.

One of the things that may drive people away from this kind of systems is their complexity. In this paper we present an e-voting protocol designed to work with e-cognocracy, much simpler than the previously existing one [1], through the use of short linkable ring signatures.

Short linkable ring signatures are a cryptographic primitive that allows one person to sign as a member of a group, but without giving any information about the identity of the signer and with no previous set up and, furthermore, all the signatures from the same signer can be linked together but keeping the anonymity. The key element they present is that, unlike other schemas, they have a constant size (making them independent of the number of people in the group).

Keywords: Short linkable ring signatures, e-voting, e-cognocracy, e-government.

1 Introduction

Democratic governments are chosen indirectly by the representatives the citizens have previously elected as a practical simplification of their participation. However, as the democratic process evolves, its flaws become more evident, and the citizenry usually loses interest and faith in their representatives as the time passes, thinking that their expectations are not going to be accurately represented. Politicians and governments have realized the problem this poses, and there have been a number of attempts to get feedback from the people

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 125–136, 2007.

(through websites, e-mail, etc.) in order to keep contact with the citizens and their interests.

Assuming that receiving opinions from the citizens is a good initiative (something that already happens when there is an electoral voting), e-cognocracy can be said to be tool to get citizens to take part in the decision making process.

In e-cognocracy [2,3], the decision making process is no longer only focused in the actual choice of the citizens, but also in the way through which this choice has been made, the actual reasons that interest and move the people (i.e. it is important to know not only the political party chosen, but whether it has been chosen because of their economical, social, etc. proposals).

Additionally, in order to collect the opinion of the people, several key issues are presented to them to vote, while still keeping the elected representatives with some weight in the choice (e.g. it could be 60% direct participation and 40% indirect participation) in order to stop demagogy.

This kind of polls is very similar to electoral voting, and thus the new added requirements must not conflict with the classical security properties of e-voting. The main new requirement is the linkability of votes, which will allow to track the evolution of the choices of the citizens individually while still keeping their anonimity.

This has led to the creation of a new cryptographic scenario which takes this goal into account and proposes a variation to classical e-voting systems. However, one of the risks faced by such proposals is that the increased complexity might drive away the citizens, generating a lack of trust derived from the necessity of letting other people verify the systems [4].

In this paper we propose an algorithm to improve the trust and perceived security in e-cognocracy modifying the previous proposal [1], simplifying the process and decreasing the number of actors involved. By using short linkable ring signatures we eliminate the need for an Electoral Authority without losing any of the services provided. We also reduce the computational complexity of the linking votes and of the receipt generation.

Furthermore, the vote is sent directly from the voter to the recount authority, giving the voter more confidence in the process (previously the voter sent his vote to an Electoral Authority which in turn sent it to the Recount Authority, and it was possible for those two entities to work together to break anonymity, even if it is a corner case).

We additionally provide a mean to mark the voters as pertaining to different groups, which is useful for e-cognocracy, because, as we said, there can be a mix of direct and indirect participation.

The structure of this paper is the following. Firstly, in Section 2 we introduce the state of the art. Then, in section 3 we will describe the protocol for the different participants in the process. In section 4 we will show how we can satisfy all the requirements needed for e-cognocracy. Additionally, an example of an actual implementation is presented in section 5. Finally, in section 6 we will present the conclusions of this paper.

2 Previous Work

2.1 e-Cognocracy

As we have said previously, one of the ways through which e-cognocracy tries to get people involved is the identification of the issues that are critical in the decision making process. This is achieved in two ways:

Multicriteria framework: Citizens are no longer asked their preferences, but how they value each option according to several criteria, and how important these criteria are for them.

Linkability of votes: Each poll is divided in several rounds (the number of rounds is fixed beforehand, and each voter can cast each votes in as many rounds as he wants, but only once each round. The last vote cast by each voter (independently of the round in which it was cast) is the one taken into account for the final result. However, all the votes from the same voter are linked together (keeping, of course, the anonymity of the voter). This allows us to track very accurately the shifts in opinion and link them to external events, thus knowing what is really driving people's opinion.

This is done adding a field to the encrypted vote so that at the decryption process it can be linked to the other votes cast by the same individual.

In another stage of the process, there can be a search for correlations between important events and changes in the voting. This can be either in a controlled environment (e.g. forums, blogs or other online discussion media). If we suppose an scenario with three parties, A, B and C, without linkability we could know that party A lost some votes, party B kept more or less the same, and party C won some. But with this property we can exactly know whether the votes went from A to B and from B to C or if they went straight from A to C.

2.2 Short Linkable Ring Signatures

Group signatures [5] allow a person to sign a message proving himself to be a member of a group, but without revealing this identity. However, there is a group manager who issues the keys to the members, and who is able to revoke the anonymity of the signer.

Ring signatures [6] are similar to group signatures in that allow a person to sign as a member of a group while keeping his identity secret. However, there is no previous setup needed and the anonymity is unconditional.

Linkable ring signatures [7] are a step further and allow to detect different signatures from the same signer without revealing his identity. However, this kind of signatures usually have the drawback of having sizes that grows linearly with the number of members in the group. This is clearly an important problem for its use in e-voting, as the size of the census can potentially be very large.

However, recently there have been works that solve this limitation [7,8,9,10]. We will use one such schema in our paper as the base for our e-voting system. The key characteristics it presents are the following:

Unforgeability: An attacker cannot generate a valid signature without actually knowing a valid pair of public and private keys.

L-Anonymity: An attacker cannot know who within the group has signed a given message.

Linkabiliy: It is possible to know whether two signatures have been issued by the same signer.

In this paper we will use the protocol defined in more detail in [8], specifically the following construction (using the notation in [11]):

$$\begin{aligned}(\text{LSR4})\ SPK\{(r,w,x,e_1,e_2) : T_1 = g^r \wedge T_2 = g^x h^r \wedge T_3 = g^{e_2} s^r \\ \wedge T_4 = wy^r \wedge T_5 = g^{e_1} t^r \wedge {T_1}^x = g^{a_1} \wedge {T_1}^{e_2} = g^{a_2} \wedge {T_4}^x = vy^{a_1} \\ \wedge {T_5}^{2e_2} g = g^x t^{2a_2} \wedge |e_1 - 2^l|, |e_2 - 2^l| < 2^\mu \wedge \tilde{y} = {g_\theta}^{e_1+e_2}\}(M)\end{aligned}$$

where $a_1 = xr, a_2 = e_2 r, w^x = v$.

In the above schema, all computations are made modulo n, where $N = pq = (2p'+1)(2q'+1)$, and p, q, p', q' are prime numbers. The parameters e_1 and e_2 are primes such that $|e_1 - 2^l|, |e_2 - 2^l| < 2^\mu$ and form the private key corresponding to the public key $2e_1e_2+1$. The public parameters include $g, h, y, t, s, g_\theta \in (\mathbb{Z}_n^*)^2$. Also, v is the accumulation of all the public keys of the members in the group.

This protocol is proven in [8] to have the mentioned properties of unforgeability, L-anonymity and linkability.

This gives a way to construct an interactive zero-knowledge proof [12] of identity. After this, we are able to construct a signature using the techniques described in [13].

The signature scheme has the following parameters:

- A number $n = pq = (2p' + 1)(2q' + 1)$ of λ bits, being p, q, p', q' prime numbers.
- The parameters l and μ chosen to ensure security against collision attacks.
- The base u for the accumulator:

$$\begin{aligned} f : (\mathbb{Z}_n^*)^2 \times \mathbb{Z}_{n/4} &\rightarrow (\mathbb{Z}_n^*)^2 \\ f : (u, x) &\rightarrow u^x \mod n \end{aligned}$$

- The parameters $g, h, y, t, s, g_\theta \in (\mathbb{Z}_n^*)^2$ without known relative logarithms.

Each pair of public and private keys is:

Private key: (e_1, e_2) distinct primes in the interval $(2^l - 2^\mu, 2^l + 2^\mu)$
Public key: $2e_1e_2 + 1$

In order to construct the signature, we first create a discrete-log relation set from the previous construction following [14,15] where the free variables r, x, e_1, e_2, a_1, a_2:

$$\begin{bmatrix}
 & g & h & y & t & s & v & T_1^{-1} & T_2^{-1} & T_3^{-1} & T_4^{-1} & T_5^{-1} & g^{-1} & g_\theta^{-1} & \tilde{y}^{-1} \\
T_1 = g^r : & r & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T_2 = g^x h^r : & x & r & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
T_1^x = g^{a_1} : & a_1 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T_3 = g^{e_2} s^r : & e_2 & 0 & 0 & 0 & r & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
T_1^{e_2} = g^{a_2} : & a_2 & 0 & 0 & 0 & 0 & 0 & e_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T_4^x = v y^{a_1} : & 0 & 0 & a_1 & 0 & 0 & 1 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 \\
T_5^{2e_2} g = g^x t^{2a_2} : & x & 0 & 0 & 2a_2 & 0 & 0 & 0 & 0 & 0 & 0 & 2e_2 & 1 & 0 & 0 \\
\tilde{y} = g_\theta^{e_1+e_2} : & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e_1+e_2 & 1
\end{bmatrix}$$

From this relation set, a non-interactive zero-knowledge proof can be easily constructed.

When a member j pertaining to a group consisting of the public keys v_i wants to sign a message M he must:

1. Calculate:
   - $w = u^{\prod_{i \neq j} v_i}$
   - $v = u^{\prod_i v_i}$.
2. Calculate the following parameters:
   - $r \xleftarrow{R} [0, \lfloor n/4 \rfloor - 1]$
   - $T_1 = g^r$
     $T_2 = g^x h^r$
     $T_3 = g^{e_2} s^r$
     $T_4 = w y^r$
     $T_5 = g^{e_1} t^r$
   - Random numbers $\hat{r}, \hat{x}, \hat{e}_1, \hat{e}_2, \hat{a}_1, \hat{a}_2$
   - $B_1 = g^{\hat{r}}$
     $B_2 = g^{\hat{x}} h^{\hat{r}}$
     $B_3 = g^{\hat{a}_1} {T_1}^{-\hat{x}}$
     $B_4 = s^{\hat{r}} g^{\hat{e}_2}$
     $B_5 = g^{\hat{a}_2} {T_1}^{-\hat{e}_2}$
     $B_6 = y^{\hat{a}_1} {T_4}^{-\hat{x}}$
     $B_7 = t^{2\hat{a}_2} g^{\hat{x}} {T_5}^{-2\hat{e}_2}$
     $B_8 = g_\theta^{(\hat{e}_1+\hat{e}_2)}$
   - $c = H(M)$ a hash of the message and a random number to avoid duplicates.
   - $\bar{r} = \hat{r} - c(r - 2^l)$
     $\bar{x} = \hat{x} - c(x - 2^l)$
     $\bar{e}_1 = \hat{e}_1 - c(e_1 - 2^l)$
     $\bar{e}_2 = \hat{e}_2 - c(e_2 - 2^l)$
     $\bar{a}_1 = \hat{a}_1 - c(a_1 - 2^l)$
     $\bar{a}_2 = \hat{a}_2 - c(a_2 - 2^l)$
   - $\tilde{y} = g_\theta^{(e_1+e_2)}$

The signature is then:

$$\sigma = \{M, \tilde{y}, T_1, T_2, T_3, T_4, T_5, B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, \bar{r}, \bar{x}, \bar{e}_1, \bar{e}_2, \bar{a}_1, \bar{a}_2\}$$

In order to verify its validity, the following shall be checked:

1. $g^{\bar{r}} = B_1 \left({T_1}^{-1} g^{(2^l)} \right)^c$
2. $g^{\bar{x}} h^{\bar{r}} = B_2 \left({T_2}^{-1} g^{(2^l)} h^{(2^l)} \right)^c$
3. $g^{\bar{a}_1} {T_1}^{-\bar{x}} = B_3 \left(g^{(2^l)} {T_1}^{-(2^l)} \right)^c$
4. $s^{\bar{r}} g^{\bar{e}_2} = B_4 \left({T_3}^{-1} s^{(2^l)} g^{(2^l)} \right)^c$
5. $g^{\bar{a}_2} {T_1}^{-\bar{e}_2} = B_5 \left(g^{(2^l)} {T_1}^{-(2^l)} \right)^c$
6. $y^{\bar{a}_1} {T_4}^{-\bar{x}} = B_6 \left(v y^{(2^l)} {T_4}^{-(2^l)} \right)^c$
7. $t^{2\bar{a}_2} g^{\bar{x}} {T_5}^{-2\bar{e}_2} = B_7 \left(g^{-1} t^{2 \cdot (2^l)} g^{(2^l)} {T_5}^{-2 \cdot (2^l)} \right)^c$
8. $g_\theta{}^{(\bar{e}_1 + \bar{e}_2)} = B_8 \left(\tilde{y} {g_\theta}^{2 \cdot (2^l)} \right)^c$.

3 Our Protocol

3.1 Actors

We start defining the parties involved in our e-voting schema, and the roles each one takes, before describing their actuation. These are:

Voter (V): Each voter must show its preferences in a multi-choice question, and rank them numerically. For all rounds of the voting, the census shall be constant.

Certification Authority (CA): The Certification Authority shall issue the public/private keys and certificates for each actor involved in the process, and will serve as Trusted Third Party with regard to the validation of certificates.

Recount Authority (RA): The Recount Authority is the only entity allowed to decrypt the votes.

3.2 System Set Up

The system set up is done only once, and in this phase the parameters common to all polls are chosen. These include:

- The signature parameters $n, l, \mu, u, g, h, y, t, s$.
- RA is given a pair of public and private keys using traditional cryptography, and publishes them in the CA.
- Each voter is given a pair of public and private keys and the public key is published in the CA.

All the parameters will be chosen randomly where appropriate.

3.3 Voting Set Up

These parameters will be different for each voting, but will be constant in all the rounds of the same voting.

– For each voting a new parameter $g_\theta \in (\mathbb{Z}_n^*)^2$ will be generated, so that the *linking-tags* of the same voter will be different across the polls and constant across the rounds of each voting.
– A census of the different groups of voters will be created and published.

3.4 Ballot Cast

When a citizen (the voter j) wants to participate in the process, they take the following steps:

1. He downloads the census of the group to which it belongs and the public keys v_i of the members.
2. He encrypts his vote M with RA's public key, obtaining M'.
3. He asks RA to start the voting process.
4. RA answers with a random challenge, for the voter to sign.
5. He signs both the challenge and H(M), being H a secure hash function, obtaining σ_c and σ respectively and sends M', σ_c and σ to RA, through an anonymous channel.
6. Upon receiving a vote, RA verifies its validity discarding it if the signature σ_c is not valid, or if the *linking-tags* present in σ_c and σ are different.
7. RA checks whether the *linking-tag* $\tilde{y}$ has already appeared in this round. If so, it discards the vote.
8. RA stores the encrypted vote M' and the clear signature σ.
9. RA attaches the current round and voting to the *linking-tag* and signs it, sending this signature back to the voter, who will be able to use it as a proof of voting.

3.5 Claiming Period

After the closing of the voting period, RA will publish a list of the *linking-tags* received, so that any voter having a valid receipt can reclaim if his vote has not been taken into account.

3.6 Tallying

Before the tallying, RA verifies again the validity of the votes, and scans for duplicates in the same round. Then the votes are decrypted with RA's public key, their signatures are verified, the ones with the same *linking-tag* are linked, and the results are published.

4 Security of the Protocol

Our system of e-voting started as a tool of e-cognocracy and it requires the following performance properties, most of which are shared with the classical systems of e-voting [16,17]:

Authentication: *Only voters in the census shall be able to vote.* The kind of signature used guarantees that the signer is in the designed group of voters.

Democracy: *Each voter shall be able to vote only once in each round.* If one voter cast two opinions in the same round, he should use the same *linking-tag* and would be detected.

Anonymity: *A voter shall not be linked to its vote.* Anonymity is guaranteed by the kind of signature used, and the probability of guessing the actual voter is $1/N$, where N is the number of voters in a given group.

No coercion: *A voter shall not be able to prove its vote.* In exchange for his voting, the voter receives only a signed *linking-tag* and time-stamp and does not bear a relation to the content of the vote. As the voter is requested to sign a random message, he is prevented from selling his vote (otherwise, the vote could be easily sold by providing both the message and its signature to a third party, who could then proceed to vote).

Accuracy: *It shall not be possible to remove a valid vote from the final counting.* Each voter signs as a member of a valid group, therefore, he must know the private key of a member, which is impossible to fake provided it has a suitable length.

Reliability: *It shall not be possible to include a non-valid vote in the final counting.* Each voter has a signature of the *linking-tag* that he or she sent to the RA and a list of these *linking-tags* will be published before the recount; therefore, even if RA is compromised, the ballots cannot be deleted, since this action will be reported by the affected voters who will present their signed *linking-tags* to support their objection.

Veracity: *Only each voter can cast its vote.* A ballot cannot be sent to RA (even if RA is compromised), because it would be necessary to obtain a valid signature, and that is not possible without the private key of the voter.

Verifiability: *Voters shall be able to verify that their vote has been correctly accounted.* For each vote received, the RA gives back to the voter a signed *linking-tag*. Later, when beginning the recount, RA publishes a list of the *linking-tags* of the ballot. If a voter has a ballot that is not included in the list, he could report it to the RA so that it undertakes the appropriate action.

Neutrality: *For each round the vote should be secret until the recount phase.* RA decrypts the votes once each round is finished.

Linkability: *Two votes from the same voter in different rounds of the voting shall be linked together, but not to the voter who cast them.* Every voter uses the same *linking-tag* in every round. We link all the ballots with the same *linking-tag* as coming from the same person.

5 An Actual Implementation of the Protocol

A workshop on e-voting and e-democracy organized by the GTC and the GDMZ was held on March 1, 2007 within the framework of the II Congress on E-Commerce arranged by the Telefónica Chair, the University of Zaragoza Communications Technology Group (GTC) and the Zaragoza City Council [18]. The

objective was to show the user the features of an ideal electronic voting system. To this end, a case study was used concerning the location, or otherwise of a NATO intelligence base in the vicinity of Zaragoza.

5.1 Description of the e-Voting Process

Introduction: The workshop began with a presentation of the e-voting process and its application in the new context of e-cognocracy. At the same time, the concepts and activities supporting the security of the process were explained. In a real situation some of these matters would be dealt with before the voting process. Nevertheless, as this was a workshop the full process was carried out. Key issues were as follows:

1. *Initialization of digital cards:* Each participant was furnished with a smart card for use as identification in the poll. However, the cards were not yet personalized in order to represent all of the phases of the process. The first phase of the workshop, then, was to initialize the cards.
2. *Ballot set up:* In order to begin a voting process, the poll must be registered in the system. This was done in a manner that was transparent for the voters. In a real-life situation both the initialization of the cards and the set up of the vote would, of course, be carried out in advance. Nevertheless, the full process was carried out in the workshop.

   When the ballot was registered in the system, the statement of the problem was explained to the participants, including information on the issue in question, the possible alternatives and the assessment criteria. In addition, instructions were given on the voting system, the use of the interface and the method to cast votes.

First Voting Round: The users then proceeded to the first round of voting, filling in their preferences and casting their votes. Each voter showed his/her preferences with several matrices. The matrices presented two-by-two comparisons between the options (criteria or alternatives), and the users were asked to weigh one option against the other. After that, the vote was encrypted, signed, and sent to RA according to the procedure described above.

Online Debate (Forum): After the first round, time was allowed for the participants to debate and discuss their opinions about the issue presented. The debate was conducted online via a forum created for the purpose. The messages published were anonymous, and each user was permitted to use a pseudonym under which to express his/her opinions.

Second Voting Round: At the end of the debate, the participants were provided with fresh information that might induce them to change their votes. The second voting round then began. While this process could have been repeated several times, and a longer period could have been left between voting rounds to allow more learning time, it was not viable to carry out more than two rounds in the context of the workshop. At the end of the second round the final results were presented, together with information on the evolution of each voter from one round to the next (while maintaining voter anonymity at all times).

5.2 Deployment Infrastructure

To end this section on the "electronic voting and e-cognocracy" workshop, let us mention that open source applications were used as tools in all cases to allow interoperability between systems. The technological characteristics of the deployment in the workshop were as follows:

1. *Hardware:* Each voting booth consisted of a desktop computer, an ACR38 smart card reader and a preprogrammed JCOP28 JavaCard smart card, in which personal details were not initialized.
2. *Software:* The software used for voting in each voting booth was:
   - M.U.S.C.L.E. Applet for JavaCard, allowing programming of the JavaCard for use to store keys and certificates.
   - Identity Alliance, which acts as middleware between the smart card software and the navigator, offering a standard PKCS11 interface.
   - Mozilla Firefox browser, capable of accessing the keys stored in the PKCS11 device.
   - JSS Module for the navigator to allow the Java applets to access the navigator's keys repository.
   - PRIOR Applet to process the electronic vote. The applet is able to access the certificates and keys contained in the navigator, and through them the keys stored in the smart card. It is divided into two clearly differentiated blocks:
     - Vote choice system: The vote choice system allows the user to issue opinions between the proposed alternatives and criteria, which will then form a part of the vote eventually cast.
     - Vote casting system: The vote casting system handles cryptography and communication with the Recount Authority. It requests and verifies the signature for the vote and sends the encrypted vote for storage.
3. *Server infrastructure:* The server used the following standard services:
   - Apache as the server for static websites.
   - Tomcat as the applications server (JSP and servlets).
   - Jk2 as a connector between Apache and Tomcat, to allow certain addresses to be transparently redirected to the applications server.
   - MySQL as the data base server.
   - OpenSSL to handle the CA's procedures.

   The software installed for the voting was as follows:
   - Voting application. This used Java Servlets and JSP pages as the interface acting as the Recount Authority.
   - Vote counting application. This used Java Servlets and both a Java applet and JSP pages as the interface. In the tallying phase it uses the Recount Authority private key to decrypt the votes and stores them. It then reads the clear votes and shows the results of the voting.

– Application for automated Certification Authority procedures. The application was created in PHP and acts as the link with OpenSSL libraries, automatically issuing and returning the certificates based on the requests received at the website. It also uses the information contained in the certificates to generate a census file.

6 Conclusions

We have implemented a secure e-voting schema which improves the one proposed in [1], removing the need for an Electoral Authority who could potentially work to game the system.

The removal of a trusted authority without losing any security services leads to an improvement in the trust and perceived security, as was shown in the workshop held in the II JCEL.

In our proposal, even if the Recount Authority were compromised, the anonimity of the voters and the integrity of the votes would no be broken so long as the channel through which the vote is cast is truly anonymous.

The workshop was an encouraging and refreshing experience, as it showed that our proposal was feasible and well-received by the public. It was as well a very useful feedback to our theoric approach.

As a future work, we are working to shorten the size of the signature, still bigger than conventional signatures, while keeping it independent of the size of the group.

References

1. Piles, J.J., Salazar, J.L., Ruíz, J., Moreno-Jiménez, J.M.: Security considerations in e-cognocracy. In: Levi, A., Savaş, E., Yenigün, H., Balcısoy, S., Saygın, Y. (eds.) ISCIS 2006. LNCS, vol. 4263, pp. 735–744. Springer, Heidelberg (2006)
2. Moreno-Jiménez, J.M., Polasek, J.M.: e-Democracy and knowledge. a multicriteria framework for the new democratic era. Journal of Multicriteria Decision Analysis 12, 163–176 (2003)
3. Moreno-Jiménez, J.M., Polasek, J.M.: e-Cognocracy and the participation of immigrants in e-governance. In: TED Conference on e-government 2005. Electronic democracy: The challenge ahead. Schriftenreihe Informatik, vol. 13, pp. 18–26. University Rudolf Trauner-Verlag (2005)
4. Riera Jorba, A., Ortega Ruíz, J.A., Brown, P.: Advanced security to enable trustworthy electronic voting. In: 3rd European conference on e-Government, pp. 377–384
5. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
6. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, Springer, Heidelberg (2001)
7. Liu, J., Wei, V., Wong, D.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 325–335. Springer, Heidelberg (2004)

8. Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. Cryptology ePrint Archive, Report 2004/281 (2004), http://eprint.iacr.org/
9. Wei, V.K.: Tracing-by-linking group signautres. Cryptology ePrint Archive, Report 2004/370 (2004), http://eprint.iacr.org/
10. Tsang, P., Wei, V., Au, M., Chan, T., Liu, J., Wong, D.: Separable linkable threshold ring signatures. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 384–398. Springer, Heidelberg (2004)
11. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 410–424. Springer, Heidelberg (1997)
12. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. 18(1), 186–208 (1989)
13. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
14. Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous identification in ad hoc groups. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 609–626. Springer, Heidelberg (2004)
15. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer, Heidelberg (2004)
16. Cranor, L.F., Cytron, R.K.: Design and implementation of a practical security-conscious electronic polling system. Technical Report WUCS-96-02, Washington University (1996)
17. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology: Proceedings of Crypto 1982, pp. 199–204 (1982)
18. Sanz, A., Moreno-Jiménez, J.M.: Taller de voto electrónico y e-cognocracia. II Jornadas de Comercio Electrónico http://voto-jcel.cps.unizar.es/

Simulation-Based Analysis of E2E Voting Systems

Olivier de Marneffe*, Olivier Pereira**, and Jean-Jacques Quisquater

Crypto Group – Université catholique de Louvain
{olivier.demarneffe,olivier.pereira,jjq}@uclouvain.be

Abstract. End-to-end auditable voting systems are expected to guarantee very interesting, and often sophisticated security properties, including correctness, privacy, fairness, receipt-freeness, ... However, for many well-known protocols, these properties have never been analyzed in a systematic way. In this paper, we investigate the use of techniques from the simulation-based security tradition for the analysis of these protocols, through a case-study on the ThreeBallot protocol.

Our analysis shows that the ThreeBallot protocol fails to emulate some natural voting functionality, reflecting the lack of election fairness guarantee from this protocol. Guided by the reasons that make our security proof fail, we propose a simple variant of the ThreeBallot protocol and show that this variant emulates our functionality.

1 Introduction

End-to-end (E2E) universally auditable voting systems, which include Punchscan [1,2], Prêt à Voter [3], ThreeBallot, VAV, and Twin [4] for instance, attract more and more attention from the scientific community due to the highly desirable security properties they can offer while preserving a high level of usability. By including the delivery of receipts that can be used for verification on a public bulletin board, those systems allow voters to get confidence into the facts that their ballot selection has been cast as intended, properly recorded, and included in the final tally. A central challenge in the design of these systems is the construction of vote receipts that can be efficiently used for auditing elections, while not introducing the possibility of coercing voters.

While these voting systems are expected to guarantee sophisticated security properties, only few works address the precise definition (and proof) of the properties guaranteed by these protocols [5,6,7,8,9]. Indeed, the fuzziness on the properties guaranteed by protocols is actually reflected in the definition of these properties: as an example, the notions of coercion resistance and/or receipt-freeness [5], which informally guarantee that a voter should not be able to convince anyone else of her vote, are defined in three different ways by Juels et al. [6], Delaune et al. [7], and Moran and Naor [8].

* Funded by the Belgian Interuniversity Attraction Pole P6/26 BCRYPT.
** Research Associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS).

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 137–149, 2007.

In this paper, we investigate the modeling and analysis of E2E voting systems through the case-study of the ThreeBallot system [4,10], which presents the interesting characteristic of not relying on the use of any computational hardness assumption. We perform our analysis by comparing the considered E2E protocol to an *ideal functionality*, following the simulation-based security approach [11,12] and the notion of secure protocol emulation(see, e.g., [13,14]).

To this purpose, we define a generic ideal functionality for voting, capturing the functionalities used by Groth [9] and Moran and Naor [8] for instance. We then describe the ThreeBallot protocol and compare it to a natural instance of our functionality. This analysis shows that the ThreeBallot protocol actually does not realize the considered functionality, reflecting the fact the election fairness in not guaranteed by this protocol (this has also been observed by Araujo et al. [15] and Clark et al. [16]). In order to reflect this limitation, we propose a relaxed version of our functionality, which quantifies the information leaked by the receipts issued in ThreeBallot election.

Observing the reasons that made the emulation proof fail for our original functionality, we then propose a simple variant of the ThreeBallot protocol and show that it securely emulates our original functionality. Relying on the composability of the secure protocol emulation notion, we split our proof into two stages: in a first stage, we show that the use of the ThreeBallot receipts guarantees the authenticity of the public bulletin board. Then, relying on this first result, we show that the ThreeBallot protocol with authentic bulletin boards emulates our functionality. We observe that, up to the forced abstention case that we did not consider here, a protocol securely emulating the functionality we consider here also satisfies the receipt-freeness property proposed by Moran and Naor [8].

2 Modeling Voting Systems

2.1 Voting in an Ideal World

Voting systems involve sophisticated protocols, designed to be used in complex environments. Specifying their properties in such context might therefore be a challenging task. A more accessible challenge consists in specifying the expected behavior of a voting system in an ideal world, where parties can be trusted, and communication channels are assumed to be private and authentic. Intuitively, in such a context, each voter V_i from a set $\{V_1, \dots V_n\}$ will simply identify herself by showing her identifier U_i to a trusted party $\mathcal{F}_{vote}$, and give this party her vote x_i. Then, when $\mathcal{F}_{vote}$ has received all votes, it will compute the election result as $f(x_1, \dots, x_n)$, and broadcast it.

This specification of a voting system probably corresponds to the first intuition we have from a voting process: all votes are as private as possible (nobody will ever be able to infer more about them than what can be inferred from the election outcome), and the election outcome is correct, as $\mathcal{F}_{vote}$ is trusted.

However, this specification is clearly too strong: we usually cannot expect a voting system to hide from external observers the mere fact that someone voted. Besides, it should probably be tolerated that an adversary can make an election

fail, by sabotaging some part of the election process or corrupting some parties. The way it is tolerated that an adversary interferes with a voting system will typically change from one system to another. Therefore, we will leave this part of the voting functionality unspecified for the moment, and simply consider that $\mathcal{F}_{vote}$ accepts to play some protocol π with an adversarial component $\mathcal{S}$, in the ideal world. (We will see concrete instances of protocol π later.) The behavior of the functionality $\mathcal{F}_{vote}$ is illustrated in Fig. 1.

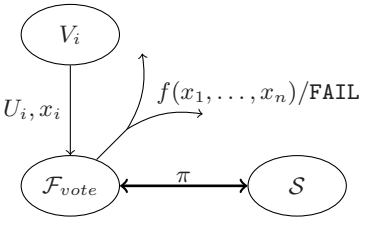

Fig. 1. Template of the ideal functionality for an E2E voting system

2.2 Voting in the Real World

We now would like to show that a real voting system securely emulates an instance of this ideal functionality. To this purpose, we use the notion of secure emulation, as proposed in the universally composable (UC) security framework [13], or in the reactive simulatability framework [14] for instance.

Intuitively, one says that a protocol emulates an ideal functionality if everything an adversary can do by interacting with the protocol can be matched by another adversary interacting with the ideal functionality. Now, if the behavior of the ideal functionality can be regarded as safe, this implies that the adversaries interacting with the protocol do not harm. The simulation-based approach of security has two main benefits: first, the ideal functionality being typically much simpler than the protocol, it is much easier to understand what a sophisticated protocol really does by looking at the functionality it emulates. Secondly, secure emulation definitions usually come with secure composition theorems, which allow using a functionality as a component of larger protocols while guaranteeing that the security properties will be preserved when the functionality is replaced by any protocol emulating it.

In the following sections we investigate the use of this secure emulation notion on a example: the ThreeBallot voting system.

3 The ThreeBallot E2E Voting System

The ThreeBallot system, originally proposed by Rivest [10] and later improved in collaboration with Smith [4], is a paper-based E2E system with the appealing specificity that it does not make use of any cryptographic algorithm or any other sophisticated computational procedure in any stage of its use. We briefly summarize the ThreeBallot system definition and the related analysis works.

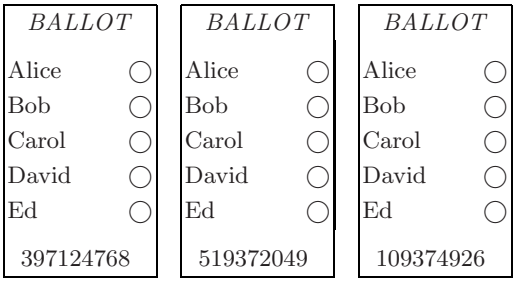

Fig. 2. An example of empty multi-ballot for a single race election with five candidates

3.1 Protocol Description

A voter taking part to an election using the ThreeBallot system receives a paper *multi-ballot* made of three ballots, each containing the list of candidates with bullets next to them, but all differing by a ballot-ID present at the bottom of each ballot. Those ballot-IDs are unique, random, all generated independently of each other. An example of multi-ballot that can be used for a race of five candidates is proposed in Fig 2.

In order to express her opinion, the voter places the three ballots side-by-side in the voting booth, and fills them as follows: (i) if she approves a candidate, she randomly fills *two* of the three bullets corresponding to that candidate, while (ii) she randomly fills *one* of the three bullets corresponding to the candidates she rejects. As a result, each row of a filled multi-ballot contains one or two filled bullets (not zero, nor three).

When this is done, the multi-ballot is inserted into a "checker machine" that verifies whether it was correctly filled. If it is the case, the voter asks the checker machine to produce a copy of one of the three ballots, which is used as a take-home receipt, and the machine then drops all three original ballots in the ballot box. One may observe that the receipt cannot be used by the voter to convince a third party of the candidate(s) for which she voted: given one ballot, it is possible to build a pair of ballots so that the reconstructed multi-ballot is a valid vote for any candidate(s) of the election.

Once the election is over, all the casted ballots are mixed and published on a Public Bulletin Board (PBB). The outcome of the election can then be computed by anyone: the number of votes for one candidate is the number of filled bullets for this candidate minus the number of voters. Each voter can also be assured that her vote was taken into account properly by checking that her receipt is on the PBB (using the ballot-ID). This verification process is expected to guarantee that, if everyone checks the integrity of its receipt on the PBB, the probability that someone can alter t ballots without being noticed is around $(2/3)^t$.

3.2 Related Works – Analysis and Critics of ThreeBallot

It has been observed and discussed by several authors that, when the number of election candidates is large, the receipt-freeness property of ThreeBallot cannot

be guaranteed. Notably, Appel [17] and Strauss [18,19] showed that a receipt associated with the PBB may result in the multi-ballot reconstruction and thus to voter privacy loss, which can lead to attacks by coercion or vote-selling. This attack is essentially possible because the number of ways of filling a ballot grows exponentially with the number of candidates, which makes it highly probable, when the number of voters is small, that only two ballots can be gathered with one given receipt to make a valid vote. Some Rivest's students at MIT showed that this kind of attack is practical [20]: they corrupted a mock election organized for a course. As a result, Rivest and Smith [4] introduced the *Short Ballot Assumption*, stating that the length of the ballots must be kept small in front of the number of voters (possibly by splitting them into several parts), which guarantees that a reconstruction is not likely to be possible. Concrete bounds on the length of the ballots have been estimated by several authors, including [19,21].

Another issue, which we address in more detail in this paper, is the fairness of the elections: Araujo et al. [15] and Clark et al. [16] show that it is possible to approximate the outcome of an election by using a few receipts only. As far as we know, no practical solution to this problem has been proposed.

Although these researches pointed issues in ThreeBallot, we are still convinced that this protocol remains quite appealing considering the simplicity of the proposed solution.

4 Modeling the ThreeBallot Protocol

We now describe our modeling of the ThreeBallot protocol, and a simple instance $\mathcal{F}_{vote}(\pi_{TB})$ of the $\mathcal{F}_{vote}$ ideal functionality, that we would expect to be emulated by this protocol. However, we will show that the ThreeBallot protocol does not emulate our candidate $\mathcal{F}_{vote}(\pi_{TB})$ functionality. Examining the reasons that make the secure emulation proof fail, we will propose a relaxed version of the $\mathcal{F}_{vote}(\pi_{TB})$ functionality, which will give a better idea of the quantity of information leaked by the ThreeBallot protocol.

4.1 Ideal World and Real World Definitions

Real World Model: Our modeling of the real world interactions taking place in the ThreeBallot system is depicted on top of Fig. 3. These interactions take place as follows. First, the adversary can send a vote x_i to each voter V_i. When voter V_i casts this vote using a (valid) $multi\text{-}ballot_i$ sent to the poling station (PS), together with the index j of the ballot she would like to have as a receipt and her identifier U_i, the polling station sends this receipt, r_i, back to V_i. It also informs the adversary $\mathcal{A}$ that this voter voted by sending him the identifier of the voter (U_i). The adversary may now ask V_i to show him her receipt through the instruction req_i. We consider that the voters answer this request every time but we will bound the number of times the adversary can issue it.

Once the casting period is over, the polling station sends the ballot box (BB) to the adversary who can modify it as he wishes in order to produce a new ballot

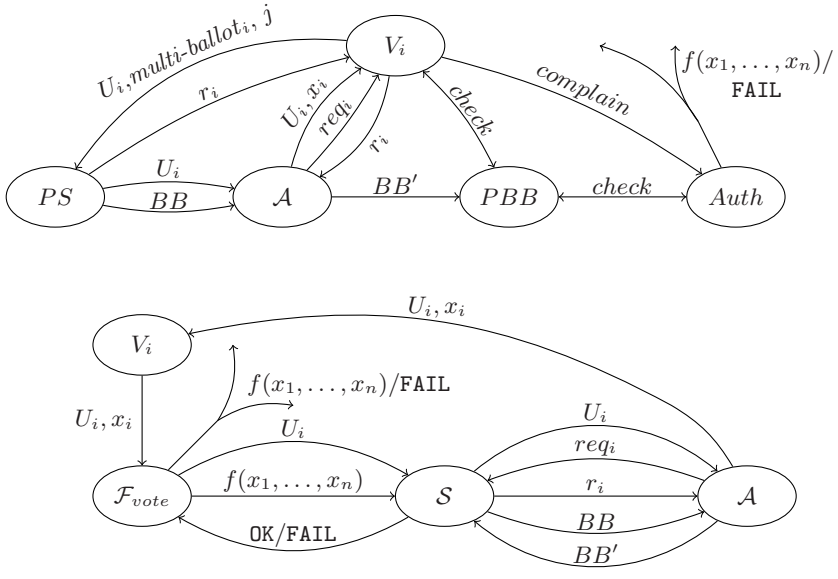

Fig. 3. Real world model (top) — Ideal world model (bottom)

box (BB') which is sent to the public bulletin board (PBB). So, we assume that the adversary is not able to make any change in the ballot box until the end of the vote casting process.

The audit process consists in several interactions between the voters, the PBB and an authority centralizing the complains. Voters check whether their receipts are present on the bulletin board and complain to the authority if it is not the case. After verification, the authority records the objections and, when the time for complaining is over, broadcasts the outcome of the election $f(x_1, \ldots, x_n)$ (that is, the number of votes casted for the different candidates), or declares that the election failed (due to tampering detected during the audit phase).

Ideal World Model: We now describe a candidate ideal functionality $\mathcal{F}_{vote}(\pi_{TB})$ for the ThreeBallot protocol, based on the $\mathcal{F}_{vote}$ functionality described in Section 2.1, and in which the π_{TB} protocol describes the interactions between the ideal voting functionality and the adversary.
$\mathcal{F}_{vote}(\pi_{TB})$ executes as follows:

1. On input x_i from voter V_i, $\mathcal{F}_{vote}(\pi_{TB})$ stores x_i and transmits V_i's identifier U_i to the adversary;
2. When the vote casting process is complete, $\mathcal{F}_{vote}(\pi_{TB})$ sends the election outcome $f(x_1, \ldots, x_n)$ to the adversary;
3. On input `OK` or `FAIL` from the adversary, $\mathcal{F}_{vote}(\pi_{TB})$ broadcasts $f(x_1, \ldots, x_n)$ or `FAIL` (respectively) as election outcome.

We observe that, in this definition, the protocol π_{TB} counts three types of messages. Messages of the first type notify the adversary that the voter V_i casted

her vote. The same information is transmitted to the adversary in our model of the ThreeBallot protocol. The second message type corresponds to the sending of the election outcome to the adversary, waiting for an approval which is sent through the last message of the π_{TB} protocol. This corresponds to the fact that, in the ThreeBallot protocol, the adversary is assumed to be able to access and change the ballot box before it appears on the bulletin board. However, the election integrity is guaranteed here: the ideal adversary is able to make the election fail, but not to make it result in a wrong tally. (Actually, small discrepancies between the ideal and real worlds will be tolerated, reflecting the fact that the ThreeBallot adversary has a noticeable probability to make small changes in the ballot box content without being caught.)

It is worth noting that, as identity of voters is considered to be public information, this ideal functionality guarantees all the classical security requirements for a voting protocol (correctness, privacy, fairness...). The only available information about the cast votes is what can be inferred from the election outcome, once it has been revealed.

4.2 Simulatability of the ThreeBallot Protocol

Following the secure emulation notion discussed in Section 2.2, we now would like to show that everything that can be done by an adversary interacting with the ThreeBallot protocol can be matched by another adversary interacting with the $\mathcal{F}_{vote}(\pi_{TB})$ functionality. To this purpose, we build a *simulator* $\mathcal{S}$ which will interact with $\mathcal{F}_{vote}(\pi_{TB})$ and use these interactions to feed $\mathcal{A}$ with a consistent view of the real world protocol execution. A high-level view of these interactions is represented in the lower part of Fig. 3.

$\mathcal{F}_{vote}(\pi_{TB})$-Based Simulation Impossibility: In order to simulate a real world protocol execution, the simulator $\mathcal{S}$ has to (i) provide $\mathcal{A}$ with the user-id of the voters, (ii) answers $\mathcal{A}$'s requests for vote receipts, (iii) send $\mathcal{A}$ a ballot-box that is going to be consistent with the election result, (iv) decide whether $\mathcal{A}$'s changes in the fake ballot box are supposed to lead to an election fail.

The first part of this simulation is trivial, as $\mathcal{F}_{vote}(\pi_{TB})$ gives $\mathcal{S}$ the user-id of the voters: $\mathcal{S}$ can just forward this message to $\mathcal{A}$. However, the second and third parts are much more problematic.

Consider, for the sake of simplicity, a single race election with only two traditional candidates, Alice and Bob. We can observe that, for each of these candidates, a voter has actually 9 ways to express her vote: these votes are depicted in Table 1.

We can see every possible ballot does not occur with the same probability. Table 2 shows the different distribution for specific election outcomes. The last line of the table gives the distribution of the receipts for any election outcome when there is a proportion p of votes for Alice. Obviously, this is also the distribution of the ballots in the ballot box.

However, the simulator may have to produce receipts during the election process: the adversary is able to ask parties for their receipts before all votes have

Table 1. Possible multi-ballots in a two candidates election

Multi-ballots for Alice	**Multi-ballots for Bob**
$\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$
$\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]$
$\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]$ $\left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$

Table 2. Distribution $\{r_i\}$ for different election outcomes – in the real world

Receipts	$r = \left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]$	$r = \left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]$	$r = \left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]$	$r = \left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$
100% "Alice"	2/9	4/9	1/9	2/9
Tie	2/9	5/18	5/18	2/9
100% "Bob"	2/9	1/9	4/9	2/9
Prop. p for Alice	$\frac{2}{9}$	$\frac{1}{9}+\frac{p}{3}$	$\frac{1}{9}+\frac{1-p}{3}$	$\frac{2}{9}$

been cast. Looking at the distributions in Tab. 2, he will however not be able to produce receipts distributed as in the ideal world if he does not know p, that is, if he does not know the election outcome in advance.

This impossibility to simulate a view of the real world election by interacting with $\mathcal{F}_{vote}(\pi_{TB})$ actually reflects the lack of election fairness guarantee of the ThreeBallot protocol: by looking at receipts, a real world adversary is actually able to obtain an estimation of the election outcome.

A Vote Leaking Version of $\mathcal{F}_{vote}(\pi_{TB})$: In order to reflect this lack of fairness in the ThreeBallot protocol, we propose a variant of our ideal functionality, which we call $\mathcal{F}_{vote}(\pi_{TBL})$, which is an instance of the $\mathcal{F}_{vote}$ functionality with a protocol π_{TBL} leaking some information on the votes casted to the adversary (while the π_{TB} protocol was keeping those votes perfectly private).

The $\mathcal{F}_{vote}(\pi_{TBL})$ functionality is essentially the same as the $\mathcal{F}_{vote}(\pi_{TB})$ functionality, except that the first part of its definition is modified as follows:

On input x_i from voter V_i, $\mathcal{F}_{vote}(\pi_{TBL})$ stores x_i and tosses a biased coin c giving "heads" with probability 4/9. Then if:

- if c is "heads", $\mathcal{F}_{vote}(\pi_{TBL})$ transmits V_i's identifier U_i to the adversary (as $\mathcal{F}_{vote}(\pi_{TB})$ does)
- if c is "tails", $\mathcal{F}_{vote}(\pi_{TBL})$ transmits U_i together with the vote x_i to $\mathcal{A}$ with probability 4/5, or with a fake vote $\overline{x_i}$ with probability 1/5.

Essentially, the $\mathcal{F}_{vote}(\pi_{TBL})$ functionality leaks vote information following the distributions described in Tab. 2: with probability 4/9, the receipt chosen by the voter will have two identical bullets, and this is independent of the cast vote (so, no information is leaked); while with probability 5/9, the receipt contains two different bullets, and these bullets give a 4/5 probability of correctly guessing

the candidate supported by the voter (so, the correct vote is transmitted with probability 4/5).

So, by interacting with $\mathcal{F}_{vote}(\pi_{TBL})$, a simulator can produce a convincing receipt r_i for V_i as follows, assuming that d_i is a random bit chosen by the simulator and b_i is the (possibly wrong) vote information transmitted by $\mathcal{F}_{vote}(\pi_{TBL})$:

$$\begin{array}{ll} \text{when receiving } (V_i) & \text{if } d_i = 0, \text{ choose } r_i = \left[\begin{smallmatrix}\circ\\ \circ\end{smallmatrix}\right] \text{ else } r_i = \left[\begin{smallmatrix}\bullet\\ \bullet\end{smallmatrix}\right] \\ \text{when receiving } (V_i, b_i) & \text{if } b_i = \text{Alice}, \text{ choose } r_i = \left[\begin{smallmatrix}\bullet\\ \circ\end{smallmatrix}\right] \text{ else } r_i = \left[\begin{smallmatrix}\circ\\ \bullet\end{smallmatrix}\right]. \end{array}$$

5 A Tweak on the ThreeBallot Protocol

In the previous section, we showed that the ThreeBallot protocol does not securely emulate the $\mathcal{F}_{vote}(\pi_{TB})$ functionality. Essentially, the reason of this non-emulation comes from the fact that the receipt of each voter contains some probabilistic information about the way she voted, which in turn prevents a simulator to produce a convincing receipt if he has no information about the content of the vote.

So, in order to securely emulate the $\mathcal{F}_{vote}(\pi_{TB})$ functionality, a protocol should not leak any *noticeable* information about the votes. This suggests a simple modification in the ThreeBallot protocol, which we present and analyze in this section.

5.1 Definition of the Modified Protocol

A simple way to modify the original protocol in order to make its execution simulatable is to choose the receipt before expressing any voting preference: therefore, no information about the vote can be leaked by any receipt. The protocol variant we suggest is as follows:

(i) While in the voting booth, the voter first randomly fills one bullet per row of the multi-ballot (or receives such a pre-filled multi-ballot);
(ii) She decides which of the three ballots she wants to be copied and taken away as a receipt;
(iii) She casts her vote as in the original protocol, except that she is no more allowed to modify the ballot she pointed out as the receipt.

It is worth noting that she is still able to express her vote by filling one bullet on the row(s) she wants to vote for on the two remaining ballots.

The verification that the receipt part is not modified by the voter can, for example, be achieved by committing on the multi-ballot to the checker machine before stage (iii), and by requiring the checker machine to verify that the receipt is not modified during that stage.

5.2 Simulatability of the Modified ThreeBallot Protocol

Claim – Our variant of the ThreeBallot protocol securely emulates the $\mathcal{F}_{vote}(\pi_{TB})$ functionality, under the following assumptions:

– the Short Ballot Assumption is satisfied, and
– the number of receipts known by the adversary is small in front of the number of voters.

Essentially, the Short Ballot Assumption guarantees that the adversary cannot reconstruct multi-ballots from the receipt he obtains. The requirement on the small number of known receipts guarantees that the adversary will not be able to select non-receipt ballots from the BB with high probability. Those two assumptions will be used to assert the ability of the simulator we propose to provide a convincing view to the adversary. We do not quantify here the different election parameters that are required for secure election; partial results in this direction can be found in [19,21].

We split our analysis into two steps, introducing an intermediate system IS between the real world system RWS and the ideal world system IWS depicted in Fig. 3.

The Intermediate System: Our intermediate system IS, which is depicted in Fig. 4, differs from RWS by the fact that the ballot box used in the tally is guaranteed to be authentic. In order to reflect this, we assume the existence of an authentic transmission functionality $\mathcal{F}_{auth}$, inspired from Canneti's message authentication functionality [13, Section 6.2], which behaves as follows: when it receives a ballot box, it forwards it to the adversary, then waits for an `OK` or `FAIL` message and, according to the value of this message, forwards the ballot box or the `FAIL` signal (respectively) to the authority.

IS is then built from RWS as follows: the polling station PS remains identical; each voter V_i is replaced by a voter V_i' that behaves as V_i except that it does not take part into any audit phase, the PBB is removed (it is not needed anymore as election integrity is now guaranteed), and the authority $Auth$ is replaced by an authority $Auth'$ that simply receives the ballot box or the `FAIL` signal and broadcasts the election outcome accordingly. The IS adversary is now built as a copy of the RWS adversary $\mathcal{A}$ interacting with a simulator $\mathcal{S}'$.

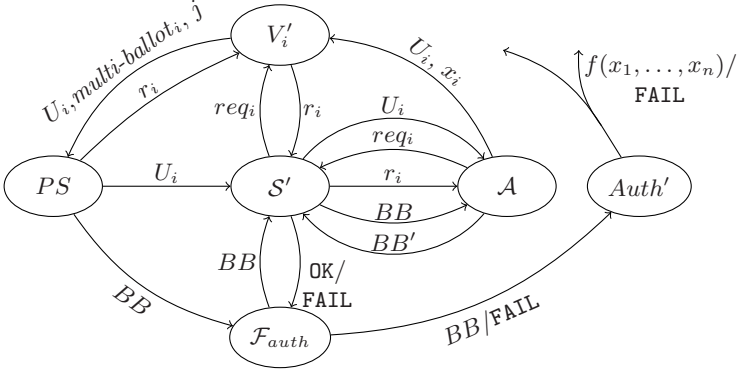

Fig. 4. Intermediate System (IS) for ThreeBallot

Table 3. Distribution of receipts in the real world (modified protocol)

Receipts	$r = \left[\begin{smallmatrix}\circ\\\circ\end{smallmatrix}\right]$	$r = \left[\begin{smallmatrix}\bullet\\\circ\end{smallmatrix}\right]$	$r = \left[\begin{smallmatrix}\circ\\\bullet\end{smallmatrix}\right]$	$r = \left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$
Probability	$\frac{4}{9}$	$\frac{2}{9}$	$\frac{2}{9}$	$\frac{1}{9}$

From RWS to IS: The distinction from the the view of $\mathcal{A}$ in RWS and in IS can come from one single place: the election outcome. Indeed, $\mathcal{S}'$ can see all other messages that $\mathcal{A}$ sends or receives in RWS, and can just forward them transparently.

So, the challenge for $\mathcal{S}'$ is essentially to decide when it must send the signal `OK` or `FAIL` to $\mathcal{F}_{auth}$. $\mathcal{S}'$ can adopt the following strategy:

- If the ballot box BB' differs from BB by a ballot that $\mathcal{S}'$ forwarded between a voter and $\mathcal{A}$, then send the `FAIL` instruction to $\mathcal{F}_{auth}$;
- If any other ballot has been modified by $\mathcal{A}$, then estimate the probability that this modification would be detected in RWS, and send the `OK`/`FAIL` message to $\mathcal{F}_{auth}$ accordingly.

The probability that any change in the ballot box will be noticed can be estimated from the distribution of the receipts in RWS, which is given in Tab. 3.

The behavior we just described guarantees that the election will fail with the same probability in RWS and IS. On the other hand, when the election succeeds, discrepancies may appear between the election outcomes in these two worlds, as the outcome in IS does not reflect the changes in the ballot box performed by $\mathcal{A}$. However, as we pointed before, if the number of voters is large, and the number of receipts obtained by the adversary is small, the difference between the two distributions of the election outcome will be small as the adversary has no further information about how to modify the BB so that it gets unnoticed.

For instance, in the case of our two candidates election, we can observe that the adversary's best strategy is to change ballots of the form $\left[\begin{smallmatrix}\bullet\\\bullet\end{smallmatrix}\right]$. If we assume that the adversary does not see any receipt, a change in such a ballot will only be noticed with probability $1/9$. Therefore, if the adversary tampers t of those receipts, he will get caught during the audit process with probability $1-(8/9)^t$ which is above $1/2$ for $t > 6$ and around $1 - 10^{-5}$ for $t = 100$ (we expect that the strong impact of a single modification notice will convince him to preserve the ballot box integrity.)

So, we can consider that the audit process of the ThreeBallot guarantees that the tally of a ThreeBallot election will be very close from the correct one.

From IS to IWS: We just showed that everything that an adversary can do by interacting with the real world system can essentially be done by an adversary interacting with our intermediate system. We now show that this behavior can also be matched by an adversary interacting with the ideal world system.

The main difference between IS and IWS lies in the fact that the IWS adversary does not see the real world receipts and ballot box anymore. Instead, he only sees the election tally before it is made public. So, we need to adapt the

strategy of our simulator accordingly, transforming $\mathcal{S}'$ into the IWS simulator $\mathcal{S}$ depicted in Fig. 3.

The strategy of $\mathcal{S}$ in order to produce the receipts and the ballot box can be as follows:

- When $\mathcal{A}$ asks for a receipt from the voter V_i, $\mathcal{S}$ produces a receipt by selecting a ballot randomly according to the receipt distribution indicated in Tab. 3;
- When $\mathcal{A}$ receives the tally from $\mathcal{F}_{vote}(\pi_{TB})$, it produces a ballot box by internally playing an election leading to the same tally, while being consistent with the already produced receipts.

The rest of $\mathcal{S}$'s behavior can be taken from the behavior of $\mathcal{S}'$.

We observe that the distribution of the receipts sent by $\mathcal{S}$ to $\mathcal{A}$ perfectly matches the one of the receipts transmitted in RWS and IS. Besides, the Short Ballot Assumption guarantees that the adversary is not able to reconstruct the multi-ballot corresponding to a receipt (linked to a voter) by finding the two missing parts in the ballot box. Therefore, $\mathcal{A}$ will not be able to distinguish the distribution of the ballot box he receives in IWS from the one he receives in the other systems. This shows that our variant of the ThreeBallot protocol emulates the $\mathcal{F}_{vote}(\pi_{TB})$ functionality and concludes our analysis.

6 Conclusion

Throughout this paper, we reported a case-study on the use of standard cryptographic protocol analysis techniques, namely simulation-based security in the line of the UC framework, on the ThreeBallot end-to-end voting protocol.

These techniques provided a systematic way to detect potential issues in the ThreeBallot scheme, and suggested ways to avoid those issues: we propose a variant of ThreeBallot that guarantees election fairness at the price of a linear loss in the precision of the election tally.

References

1. Fisher, K., Carback, R., Sherman, A.T.: Punchscan: Introduction and System Definition of a High-Integrity Election System. In: Ryan, P. (ed.) IAVoSS Workshop On Trustworthy Elections (WOTE 2006) (2006)
2. Popoveniuc, S., Hosp, B.: An introduction to punchscan. In: Ryan, P. (ed.) IAVoSS Workshop On Trustworthy Elections (WOTE 2006) (2006)
3. Chaum, D., Ryan, P.Y.A., Schneider, S.A.: A practical, voter-verifiable election scheme. Technical Report CS-TR: 880, School of Computing Science, Newcastle University (2004)
4. Rivest, R.L., Smith, W.D.: ThreeVotingProtocols: ThreeBallot, VAV, and Twin. In: Electronic Voting Technology Workshop (EVT 2007) (2007)
5. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections. In: Proceedings of the 26th ACM Symposium on Theory of Computing, may 1994, pp. 544–553. ACM Press, New York (1994)
6. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: ACM Workshop on Privacy in the Electronic Society, pp. 61–70. ACM Press, New York (2005)

7. Delaune, S., Kremer, S., Ryan, M.: Coercion-resistance and receipt-freeness in electronic voting. In: 19th IEEE Computer Security Foundations Workshop (CSFW-19), pp. 28–42. IEEE Computer Society Press, Los Alamitos (2006)
8. Moran, T., Naor, M.: Receipt-free universally-verifiable voting with everlasting privacy. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 373–392. Springer, Heidelberg (2006)
9. Groth, J.: Evaluating security of voting schemes in the universal composability framework. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 46–60. Springer, Heidelberg (2004)
10. Rivest, R.L.: The ThreeBallot voting system. Accessed on (August 12, 2007) (2006), Available from http://people.csail.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf
11. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. In: Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC 1985), pp. 291–304 (1985)
12. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game a completeness theorem for protocols with honest majority. In: Proceedings of the 19th Annual ACM Symposium on the Theory of Computing (STOC), pp. 218–229. ACM Press, New York (1987)
13. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Naor, M. (ed.) Proceedings of the 42nd Annual Symposium on Foundations of Computer Science, pp. 136–145. IEEE Computer Society Press, Los Alamitos (2001), Full version available on http://eprint.iacr.org/2000/067
14. Pfitzmann, B., Waidner, M.: A model for asynchronous reactive systems and its application to secure message transmission. In: IEEE Symposium on Security and Privacy, Oakland, CA, May 2001, pp. 184–200. IEEE Computer Society Press, Los Alamitos (2001)
15. Araujo, R., Custodio, R.F., Graaf, J.v.d.: A verifiable voting protocol based on Farnel. In: Benaloh, J. (ed.) IAVoSS Workshop On Trustworthy Elections (WOTE 2007) (2007)
16. Clark, J., Essex, A., Adams, C.: On the security of ballot receipts in e2e voting systems. In: Benaloh, J. (ed.) IAVoSS Workshop On Trustworthy Elections (WOTE 2007) (2007)
17. Appel, A.A.: How to defeat rivest's threeballot voting system. Accessed on August 12, 2007 (2006), Available from http://www.cs.princeton.edu/∼appel/papers/DefeatingThreeBallot.pdf
18. Strauss, C.: The trouble with triples. a critical review of the triple ballot (3ballot) scheme. Accessed on August 12, 2007 (2006), Available from http://www.cs.princeton.edu/∼appel/voting/Strauss-TroubleWithTriples.pdf
19. Strauss, C.: A critical review of the triple ballot voting system, part2: Cracking the triple ballot encryption. Accessed on August 12, 2007 (2006), Available from http://www.cs.princeton.edu/∼appel/voting/Strauss-ThreeBallotCritique2v1.5.pdf
20. Jones, H., Juang, J., Belote, G.: Threeballot in the field. Term paper for MIT course 6.857. Accessed on August 12, 2007 (2006), Available from http://theory.csail.mit.edu/classes/6.857/projects/threeBallotPaper.pdf
21. Henry, K., Stinson, D.R., Sui, J.: The effectiveness of receipt-based attacks on threeballot. Cryptology ePrint Archive, Report 2007/287 (2007), http://eprint.iacr.org/

A Simple Technique for Safely Using Punchscan and Prêt à Voter in Mail-In Elections

Stefan Popoveniuc[1] and David Lundin[2]

[1] George Washington University, Washington DC, USA
poste@gwu.edu
[2] University of Surrey, Guildford, Surrey, UK
d.lundin@surrey.ac.uk

Abstract. We apply a technique inspired by Scantegrity to Punchscan and Prêt à Voter and show how this results in a mail-in ballot system that is auditable, simple to use and easy to understand.

1 Introduction

Mail-in ballots are becoming increasingly common around the world, in constituencies large and small. The motivation for this include allowing those unable to visit the polling station (such as those in the military stationed overseas; expatriates; those less abled) to cast votes from a remote location and to widen participation.

The normal procedure for mail-in (or "postal") ballots is that the materials (including ballot forms and a number of envelopes) are sent to registered voters, who have requested it, in the post. The voter is instructed to fill out the ballot form in private and to place it in an envelope. This envelope is then put in another envelope on which the identity of the voter is printed: the voter signs this envelope. This multi-layered document is subsequently posted, in yet another envelope, to the election authority.

At the election authority a number of clerks open the outermost envelopes of the received ballots. This reveals the identity of the voter who has posted the ballot and when the clerks have checked that she is eligible, the signed envelope is opened and the contents dropped into a ballot box. When all ballots have been dropped into the ballot box this is shaken so that it is infeasible for anyone to guess the identity of the posting voter. The contents of this ballot box can now be tallied together with ballots collected in polling stations.

The most obvious flaw in this process is the fact that the voter is not offered the secrecy of a voting booth. We consider a voting booth one of the very few methods of ensuring that voters are able to cast secret ballots. When sending a ballot from home, a coercer, someone in the home or an external vote buyer for example, may require to be present when the ballot form is filled out and there appears to be no way to stop this. As with traditional paper-based voting schemes the voter must also place trust in people and procedures to ensure that the election is fair and accurate.

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 150–155, 2007.

The description of the mail-in ballot procedure described earlier may arguably be a best-case scenario — many places offer, in our view, even less secrecy and security.

Some recently suggested electronic voting schemes that use paper based ballot forms, such as Prêt à Voter [1] and Punchscan [5,4], use a voting booth as a mechanism to safeguard election secrecy [6,7]. However, it seems immediately clear that as these ballot forms do consist of paper it is possible to send them in the post, delivering the same level of secrecy (based on trust in people and processes) as traditional mail-in ballots.

In this paper we explore this notion further and propose a method whereby paper based electronic voting systems with encrypted receipts may be used in mail-in elections and offer added levels of security and proof of accuracy.

2 Techniques That Do Not Work

It has been proposed that both Punchscan [4] and Prêt à Voter [1] are able to handle mail-in ballots in their current guise. However, in this section we go through a number of configurations of both systems and show why they are not as secure as immediately thought.

2.1 Punchscan

In the case of Punchscan the immediate technique to handle mail-in ballots is to allow the voter to mark her two-page form in the comfort of her own home using the required dauber. The voter then separates the two pages, chooses one and posts this to the election authority. This page is scanned but kept secret by the authority. After the close of the election a meeting of the election officials generates not the page that the voter has posted to the authority but its twin. In short, the page published on the web bulletin board will be the same as the one the voter has still got.

It seems that this will provide the voter with a way to verify the inclusion of her vote in the tally, just as with those votes collected in polling stations. However, in polling stations the page that the voter is going to keep is scanned and immediately published onto the web bulletin board. In the mail-in ballot case the other page is scanned and the first is generated from this. This means that if the voter deliberately marks the two pages separately, the "receipt" will not match the page published on the web bulletin board. As there is no way of detecting whether this is because the ballot form has been maliciously altered by the authority or the voter has marked the pages differently, the page kept by the voter cannot be regarded as a useful receipt.

Furthermore, there are issues with the chain of custody of the page that has been posted in by the voter. When the authority receives the page a clerk will take it from its envelope and scan it. The clerk, who has seen this page, will be able to tell from the other page, published on the web bulletin board, how the voter has voted.

2.2 Prêt à Voter

It was proposed quite early [1] that Prêt à Voter is able to handle mail-in ballots simply by allowing the voter to fill out the normal ballot form in her home, tear along the perforation and then post the right-hand part to the election authority. This has the advantage over traditional mail-in ballots that the vote is encrypted when it reaches the election authority and the receiving clerk thus has no way of discerning the voter's choice(s). However, this does not offer the voter a receipt. Furthermore, it weakens coercion resistance even further as the destruction of the left-hand side of the ballot form, the randomised candidate list, cannot be enforced. Those voters who are able and willing to safeguard the secrecy of their vote will fill out the form in secret and, as soon as possible thereafter, destroy the candidate list — but some may be unable to do so for reasons already explained.

In order to provide the voter with a receipt that she can subsequently use to check the inclusion of her vote in the tally, this scheme may be extended with a carbon paper which is overlaid the right-hand side of the ballot form. When the voter makes her marks on the ballot form these marks are thus made on two overlain sheets of papers. When she has made her marks and detached and destroyed the left-hand side of the form what remains are two copies of the encrypted receipts. The voter posts the original and keeps the carbon copy. It is quite clear that a malicious voter[1] could mark the original and the carbon copy of the encrypted receipt separately and thus achieve the same attack as described in the previous section.

A further extension to address the destruction of the candidate list is as follows. When the voter has marked her form she detaches the carbon copy and keeps this for her own record. She then tears along the perforation and separates the two sides of the original ballot form. The left-hand side, the one carrying the names of the candidates, is placed in an envelope which has a window through which a serial number printed underneath the candidate list is visible. The voter then places this envelope together with the encrypted receipt in another envelope which has her identity printed onto it. She signs this envelope and posts it to the election authority.

When such a mail-in ballot is received the clerk checks that the voter is eligible and registered and if so, it is checked, without opening the envelope, that the candidate list posted in by the voter is the one that matches the encrypted receipt. If this check is passed then the candidate list is shredded without being taken out of the envelope and the encrypted receipt is scanned and submitted to the web bulletin board. However, the voter might attack this solution by placing not the candidate list but another piece of paper onto which the serial number has been printed into the innermost envelope.

3 Our Technique

In a recent proposal by David Chaum the Scantegrity system [2] seeks to add a certain level of verifiability to all paper-based election systems, be it votes

[1] The authors would like to note that the term *malicious voter* is a favourite.

that are collected in a polling station and scanned (immediately or in a central location) or mail-in ballots. In essence the system is based on a chit, a corner of the ballot form, which holds the serial number of the form and is torn from it by the voter before the form is submitted. At the time of casting the voter makes a note of the randomly assigned letter that represents the choice she has made. After the close of the election the authority publishes the serial numbers of all ballot forms used together with the letters chosen by the voters. This does not leak any information about the contents of the vote but the voter can check that the published letter matches her note. If it does not, the voter can challenge the election through a procedure which proves that the chit matches the ballot form and that the correct letter has been marked on that form, without revealing the contents of the vote. Please refer to [3] for details of this system.

The Scantegrity approach immediately works with mail-in ballots and we simply propose the addition of the chit to both the Punchscan and Prêt à Voter ballot forms that are to be used as mail-in ballots. All the actions the voter must undertake in order to vote by mail-in ballot are thus as follows:

The voter marks her choices on the ballot form and separates it into two, by detaching the pages or the columns from each other. In the case of Punchscan she randomly selects one page to keep — her receipt. She then detaches the chit from the receipt and puts this in a safe place, along with a note of the choice(s) she has made at each position on the form. The part of the form that must be destroyed is placed into an envelope that has a window through which the serial number of this part can be read.

Along with this envelope the encrypted receipt is placed in yet another envelope which bears the voter's identity. Having sealed this envelope the voter signs it to indicate to the election authority that she has cast her vote in accordance with the rules and so forth. With this signature she might also assure the authority that she has filled the form out in private and that she has not been asked to vote in a particular way by anyone — this, naturally, is part of election law of each particular constituency. This envelope is placed in an envelope which is pre-printed with the address of the receiving election authority and when this has been sealed it can be dropped in the nearest post box.

When the election authority receives the ballot, its outermost envelope is opened by a clerk who verifies the identity of the voter and her signature. If this check passes then the envelope is stripped off. The next check that must be performed is that the serial number of the part of the ballot form that has been placed in the innermost envelope matches the serial number of the encrypted receipt. This procedure is necessary to ensure that the voter returns the part of the ballot form that must be destroyed. In order to make this as secure as possible the election authority has placed a rubber stamp onto the serial number of both parts of the form. If the serial numbers match, then the clerk shreds the envelope containing the discarded part of the ballot form and scans the encrypted receipt, causing it to be posted to the web bulletin board. The original encrypted receipt is then filed according to its serial number so that it may be retrieved with relative ease at some future point.

The voter is now able to visit the web bulletin board and call up her encrypted receipt, using the serial number printed on her chit. She can compare this to her notes and if they correspond she can trust that her vote is included in the tally.

In the case where the voter finds that the online representation of her encrypted receipt does not match the notes she made at the time of voting, she may use her chit to verify that her receipt was scanned correctly. She, or her representative (it may be necessary to limit the number of such checks as this can be done without compromising the correctness of the audit of the election) can take the chit to the election authority. Under the scrutiny of independent, or partisan, auditors and media coverage the election authority can retrieve the encrypted receipt from the archive and show, perhaps using a microscope or any level of forensic equipment required by the auditors, that the chit and encrypted receipt were once one piece of paper.

When it has been shown that the chit matches the encrypted receipt the auditors can check that the contents of the encrypted receipt does match its electronic representation published on the web bulletin board. In Scantegrity, which uses plain-text ballot forms, this check is rather tricky, please see [2,3] for details. However, in the case of Prêt à Voter and Punchscan, where the receipt is encrypted, it is safe simply to show the receipt to the voter or any representative that she might have mandated.

We have thus modified the Punchscan and Prêt à Voter voting procedure by adding Scantegrity. Instead of keeping her actual encrypted receipt and checking that it is represented correctly on the web bulletin board the voter is able to challenge the election and have this checked in any case where she may realise that there may have been errors in the scanning process — or someone has maliciously changed her vote.

4 Further Development

As we are proposing a set of procedural safeguards for the mail-in ballots in Prêt à Voter and Punchscan, it is interesting to note that further procedural safeguards can be added in order to make the duties of the voter simpler and as reliable as possible. For example, if each part of the ballot form has an edge that is cut in a way that is different from the other part, then the envelope can be designed only to reasonably hold the required part of the form. Alternatively, each part of the form may be coloured differently and the colour of each envelope correspond to this.

5 Discussion

As we feel that mail-in ballots are inherently less safe than ballots collected in a polling station, and in view of what we perceive to be an ever growing desire to use mail-in ballots, our starting point for this work has been to make this process as safe as possible. We believe we have achieved some progress toward this goal by forcing the destruction of the appropriate parts of the ballot form. It

also seems that although the reliability of the election system rests on a number of procedures we have been able to remove some potential leaks of information which may compromise the secrecy of the election.

However, it is immediately clear that we have been unable to solve the most serious problem with mail-in ballots, namely the inability to protect the voter, and even the electoral system, from coercers who may have a completely different level of access to a large number of private homes than it may to voting booths that are under the scrutiny of observers.

6 Summary

As electronic voting is becoming mature it is important to remember an old set-up that has allowed voters who have been unable to attend the polling station during election day to take part in the democratic process, namely the mail-in (postal) ballot. Arguably the greatest drawback of mail-in ballots is that the amount of trust that voters must place in people and processes (not to mention the postal service). Using a very recent concept, Scantegrity, in the paper based electronic voting schemes Prêt à Voter and Punchscan, we have here introduced ways where the mail-in ballot system becomes practicably and safely auditable.

References

1. Chaum, D., Ryan, P.Y.A., Schneider, S.: A Practical, Voter-Verifiable Election Scheme. In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, Springer, Heidelberg (2005)
2. Chaum, D., et al.: Scantegrity.org (2007), http://www.scantegrity.org
3. Chaum, D., et al.: The Scantegrity System, An Introductory Whitepaper and Example, http://www.scantegrity.org
4. Fisher, K., Carback, R., Sherman, T.: Punchscan: Introduction and System Definition of a High-Integrity Election System. In: IAVoSS Workshop On Trustworthy Elections (WOTE 2006), Cambridge, UK (2006)
5. Popoveniuc, S., Hosp, B.: An Introduction to Punchscan. In: IAVoSS Workshop On Trustworthy Elections (WOTE 2006), Cambridge, UK (2006)
6. Ryan, P.Y.A., Peacock, T.: Prêt à Voter: A Systems Perspective, Technical Report University of Newcastle, CS-TR: 929 (2005)
7. Ryan, P.Y.A., Peacock, T.: A Threat Analysis of Prêt à Voter. In: IAVoSS Workshop On Trustworthy Elections (WOTE 2006), Cambridge, UK (2006)

Threat Analysis of a Practical Voting Scheme with Receipts

Sebastien Foulle[1], Steve Schneider[2], Jacques Traoré[1], and Zhe Xia[2]

[1] France Telecom, Orange Labs
42 rue des Coutures, 14066 Caen Cedex, France
{sfoulle.ext,jacques.traore}@orange-ftgroup.com
[2] Department of Computing, University of Surrey
Guildford, Surrey, GU2 7XH, UK
{s.schneider,z.xia}@surrey.ac.uk

Abstract. Kutylowski et al. have introduced a voter-verifiable electronic voting scheme "a practical voting scheme with receipts", which provides each voter with a receipt. The voter can use her receipt to check whether her vote has been properly counted in the final tally, but she cannot use the receipt to prove others how she has voted. Another interesting property of this scheme is that, thanks to the repetitive robustness mix network, the ballot tallying phase only needs to be audited if the final results fail to achieve some conditions. However, we will show that this scheme is vulnerable to some threats, adversaries can not only violate voter privacy, but also forge the election result.

Keywords: Voter verifiable, electronic voting, threat analysis, voting ducks.

1 Introduction

Recently, voter-verifiable e-voting schemes have attracted a lot of interest. These schemes guarantee voter privacy and meanwhile achieve end-to-end verifiability. The first prototype was introduced by Chaum in [3], followed by Neff's scheme [10], [9], the Prêt à Voter schemes [4], [12], the Scratch & Vote scheme [2], Punchscan [1], etc.

In voter-verifiable election schemes, voters need to cast their votes in a secure place, e.g. in a voting booth, where the ballot form is either printed on a paper or displayed on the DRE screen. Because the whole ballot form is available to the voter, she knows how she has voted. After that, some part of the ballot form needs to be destroyed and the voter can keep the remaining part as the receipt. The voter can use the receipt to verify that her vote has been correctly counted, but she cannot use the receipt to prove others how she has voted, therefore preventing coercion, intimidation and ballot selling. Furthermore, the accuracy of the final result in these schemes is mainly relying on some hard problems, therefore voters do not need to trust the participants who run the election or equipment used in the election.

The scheme introduced by Kutylowski et al. in [7] is one of these voter-verifiable schemes. But compared to the other schemes, it enjoys two special properties:

– Thanks to the repetitive robustness mix networks, the ballot tallying phase only needs to be audited if the final result fail to achieve some conditions, therefore, Randomised Partial Checking mechanism [5] or Neff's mix [8] can be eliminated.

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 156–165, 2007.

– The verification process is much simpler, ordinary voters can verify that their votes have been counted without special knowledge. If all votes are properly recorded and tallied, voters will see that some information in their receipts (the identifiers) are correctly displayed on the bulletin board in the final result. Otherwise, their receipts can be used as the proof to make accusation.

In this paper, we will first briefly review the election scheme introduced in [7], then the scheme will be analysed using some threat based system perspectives. We will show that the scheme is vulnerable to a number of attacks. Because of some technical drawbacks or improper implementation, adversaries can not only violate voter privacy, but also forge the election result.

2 The Voting Scheme

2.1 An Overview of the Scheme

1. **Ballot construction:** when the voter enters the voting booth, the voting machine will generate two ballots and displays them on the screen, one row per ballot. Each ballot contains the encryptions (*Onions*) of the candidate names and an identifier in a random order. The voting machine commits to the ballots by printing hashes of the whole encrypted data on the so-called *hash ballot*, in the same order.
2. **Ballot casting and ballot checking:** the voter chooses one of the two ballots, and then she selects both the ciphertexts corresponding to her favourite candidate and the identifier from the chosen ballot. At this time, the voting machine prints these ciphertexts on the *voting ballot*. The voter can also ask the voting machine to open the second ballot, by printing a *control ballot* which contains the random data used for the encryptions on the second ballot, and the candidate names as well as the identifier, in the same order as they appear on the screen. This *control ballot* can be checked afterwards by any watch dog organisation. Thus, if a voter checks her *control ballot*, a dishonest voting machine will be caught with probability $\frac{1}{2}$.
3. **Ballot tallying:** After the end of the election day, the votes on the bulletin board are sent to a mixnet which applies several partial decryptions. This optimistic mixnet is making use of signatures embedded in the (plaintext) votes. These signatures from the voting machines prevent dishonest mixes to remove some votes or forge new ones. The result of the tally will be announced under the following two conditions:
   – There are as many tallied votes as votes on the bulletin board (in fact there will be exactly twice as much, and this is also true for the identifiers, see the following point (4));
   – All signatures for the plaintexts are valid.
4. **Voter verification:** Each voter will be convinced that her vote has not been manipulated, and it is in the final tally. Indeed, each *voting ballot* is made up with two ciphertexts of the chosen candidate and two ciphertexts of the voter identifier. After the ballot tallying phase, if no manipulation has been done, each voter can check that her identifier appears exactly twice. Moreover, the four *Onions* of the *voting ballot* cannot be distinguished by others, hence even if a mix server, for instance the first one, is able to modify the four *Onions* of a given ballot, she has only a 1-in-6

chance $\left(6 = \binom{4}{2}\right)$ to successfully cheat, that is substituting the encryptions of the candidate name while leaving the encrypted identifiers untouched. In the case of a successful manipulation, the voter cannot detect it but the backtracking procedure will find the mix server cheating.

5. **Receipt freeness:** It should be noted that this scheme is receipt-free, since the identifier is not embedded in the encryptions of the candidate name.

2.2 Some Technical Details

Encryption and decryption: Let us briefly describe the encryption scheme. Denote by G a cyclic group of prime order p with hard discrete logarithm problem, and let g be a generator of G. Suppose there are λ mix servers, and each mix server has a secret key x_j, and the corresponding public key is $y_j = g^{x_j}$. An *Onion*, which is an encryption of a message m, is given by the formula $c = (m \cdot (y_1 \cdots y_\lambda)^{k_1}, g^{k_1})$, where k_1 (modulo p) is chosen uniformly at random.

The inputs to the first mix server are couples $(\alpha, \beta) = (m \cdot (y_1 \cdots y_\lambda)^{k_1}, g^{k_1})$ and the outputs are $(\alpha_1, \beta_1) = (\alpha\beta^{-x_1}(y_2 \cdots y_\lambda)^{r_1}, \beta g^{r_1}) = (m \cdot (y_2 \cdots y_\lambda)^{k_2}, g^{k_2})$ with r_1 chosen at random and $k_2 = k_1 + r_1$.

Similarly, the inputs to the ith mix server are $(\alpha_i, \beta_i) = (m \cdot (y_i \cdots y_\lambda)^{k_i}, g^{k_i})$. And in particular, the last mix server receives $(\alpha_\lambda, \beta_\lambda) = (m \cdot {y_\lambda}^{k_\lambda}, g^{k_\lambda})$, and she can recover the original message as $m = \alpha_\lambda {\beta_\lambda}^{-x_\lambda}$.

Note that the first and the last mix server have more power than the others as the first mix server knows the relationships between voters and received votes, and the last mix server can compute all plaintext messages.

The ballot construction: Suppose there are two candidates in the election, the Blue Party and the Yellow Party, denoted as B and Y respectively. Once a voter is authorised to enter the voting booth, the voting machine will generate the following *virtual ballot* (it exists only in the voting machine's memory), with couples ordered at random:

$$r \;\; r_U \;\; ({B_1}^U, {B_2}^U) \;\; ({Y_1}^U, {Y_2}^U) \;\; ({I_1}^U, {I_2}^U)$$

$$q \;\; r_L \;\; ({Y_1}^L, {Y_2}^L) \;\; ({I_1}^L, {I_2}^L) \;\; ({B_1}^L, {B_2}^L)$$

The unique random string r is an identifier, and I is its encrypted ciphertext. The random string q is a seed for all random exponents. The random strings r_X with $X = U, L$ allow to distinguish the votes for a given candidate.

Each voting machine has two key pairs for signature schemes. One private key K' is used for signing the plaintexts using the signature scheme sig'. For each ciphertext ${Z_i}^X$, where $Z = B, Y, I, \;\; i = 1, 2$ and $X = U, L$, after decoding, if ${Z_i}^X$ is an encryption of a vote for candidate B or Y, we will get the plaintext

$$(Z, r_X, ser_V, {sig'}_{K'}(Z, r_X, i))$$

and the decryption of ${I_i}^X$ gives us the identifier

$$(r, ser_V, {sig'}_{K'}(r, i, X))$$

The serial number sev_V of the voting machine allows to know which is the public key to use in order to verify the signature ${sig'}_{K'}$.

The other private key K is used for creating seeds for the *Onion* construction. ${Z_i}^X$ is an ElGamal ciphertext built with random exponent k_1 as ${Z_i}^X = (m \cdot (y_1 \cdots y_\lambda)^{k_1}, g^{k_1})$, where $k_1 = sig_K(q, i, X, Z)$ and sig is a deterministic signature scheme.

After that, the voting machine has to creates and prints a *hash ballot*, which is the commitment to the above *virtual ballot*. Denote σ as the signature with secret key K of the whole data on the *hash ballot* and h represents a hash function. The voting machine will print the following *hash ballot*:

r	$h(r_U)$	$h({B_1}^U)$	$h({B_2}^U)$	$h({Y_1}^U)$	$h({Y_2}^U)$	$h({I_1}^U)$	$h({I_2}^U)$	$h(q)$
$h(r)$	$h(r_L)$	$h({Y_1}^L)$	$h({Y_2}^L)$	$h({I_1}^L)$	$h({I_2}^L)$	$h({B_1}^L)$	$h({B_2}^L)$	σ

Casting a vote: In the voting booth, the voting machine will display these two ballots on the screen in the same order as in the *virtual ballot*:

$$\begin{bmatrix} Blue & Yellow & Identifier \\ Yellow & Identifier & Blue \end{bmatrix}$$

Suppose this voter chooses the upper line on the screen and the Blue Party, the voting machine will print the *voting ballot* which contains two ciphertexts for the Blue Party, two ciphertexts of the identifier r, and the signature of the voting machine.

${B_1}^U$	${B_2}^U$	${I_1}^U$	${I_2}^U$	$sig_K({B_1}^U, {B_2}^U, {I_1}^U, {I_2}^U)$

The voter can also asks the voting machine to generate and print a *control ballot* for the ballot in the lower line as

r	Yellow Party		Identifier		Blue Party	
r_L	${Y_1}^L$	${Y_2}^L$	${I_1}^L$	${I_2}^L$	${B_1}^L$	${B_2}^L$
	$\sigma(1, L, Y)$	$\sigma(2, L, Y)$	$\sigma(1, L, I)$	$\sigma(2, L, I)$	$\sigma(1, L, B)$	$\sigma(2, L, B)$

where $\sigma(i, L, Z) = sig_K(q, i, L, Z)$.

In the voting booth, the signature on the *voting ballot* is verified before registration of the vote. Thus no vote on the bulletin board has been forged by dishonest voters.

Finally, the voter leaves the polling place with his three ballots (*voting ballot*, *control ballot* and *hash ballot*) and she can give them to a trusted organisation for verification. The organisation should help this voter to verify the following points:

- The signatures on all these three ballots are valid.
- The *hash ballot* contains the right commitment of the *control ballot*.
- *Onions* on the *control ballot* will be verified without the private key as:

  $$ {Z_i}^X = (m \cdot (y_1 \cdots y_\lambda)^{k_1}, g^{k_1}) $$

  where $k_1 = \sigma(i, X, Z)$, and $m = (Z, r_X, ser_V, {sig'}_{K'}(Z, r_X, i))$ if $Z = B, Y$, or $m = (r, ser_V, {sig'}_{K'}(r, i, X))$ if $Z = I$. In the *control ballot*, each party and the identifier should appear exactly once, and their order must be the same as in the *hash ballot*.

– The identifier value must be the same on the *hash ballot* and the opened *control ballot*.

The voter can also look at the bulletin board to be sure that his *voting ballot* has been accurately recorded.

Bulletin board, ballot tallying and backtracking: After the last decoding, one has to check the following points:

– All signatures ${sig'}_{K'}({Z_i}^X)$ for the plaintexts are valid.
– The number of votes counted on the bulletin board is exactly half the number of votes $(Z, r_X, ser_V, {sig'}_{K'}(Z, r_X, i))$ (here $Z = B, Y$), and the number of identifiers $(r, ser_V, {sig'}_{K'}(r, i, X))$.
– No vote or identifier has been duplicated.
– Each identifier and each triple (Z, r_X, ser_V) appears twice.

If all these conditions are satisfied, the results are announced. If at least one vote or identifier is invalid, the last mix server must indicate where it comes from, and she has to prove that she has decoded correctly using equality of discrete logarithm proof [14]. This is repeated with the preceding mix server until one finds a mix server unable to prove her correct behaviour. This procedure together with the embedded signatures prevent any manipulation from mix servers such as removing, inserting, duplicating or modifying an *Onion*.

3 Some Technical Errors

There are four minor technical errors in the description of the scheme:

1. The random string q is printed nowhere, although it is absolutely necessary to check the signature on the *control ballot*, since $\sigma(i, X, Z) = sig_K(q, i, X, Z)$. Therefore, it needs to be printed at least on the *control ballot*, it should be printed on the *hash ballot* as well.
2. The authors have not explained how to print the ciphertexts on the *voting ballot*. If they are printed as explained in the previous section, the probability of a successful vote replacement by the first mix server is not $\frac{1}{6}$, but $\frac{1}{2}$, because the first mix server can distinguish the four *Onions* into two groups, although she does not know the content for each group. The correct implementation of this point is that the voting machine has to randomly permute the ciphertexts of candidate and the ciphertexts of identifier before printing them on the *voting ballot*.
3. Even if the voting machine is honest to print the four ciphertexts in a random order on the *voting ballot*, it is still possible for the first mix server to classify the four ciphertexts into two groups if provided with the corresponding *hash ballot*, because the hashed ciphertexts in the *hash ballot* are all grouped, the first mix server can hash the ciphertexts on the *voting ballot* and compare the result with the *hash ballot*. Therefore, in this case, the first mix server still can replace the vote without changing the identifier with $\frac{1}{2}$ possibility.
4. It is not the *hash ballot* that should be marked as used, but the *control ballot* which contains opened *Onions*. We will illustrate this point in the next section.

4 Threat Analysis

Some papers [6], [11] have shown that electronic voting schemes are very complex and different attacks may be applied by adversaries. Some of these attacks are because of technical drawbacks, while others can be caused by improperly implementation of the e-voting systems. In this paper, we will classify our threats to the scheme in [7] into three categories:

- **Threats against anonymity:** These attacks only try to find out how a certain voter has casted her vote, breaking the voter-vote links. We have to note that these attacks are not equal to passive attacks, since some active attacks can also be used to violate voter privacy.
- **Threats against correctness:** The purpose of these attacks is to violate the correctness of the final result. Sometimes, the adversaries cannot forge the result as their wish, but they can make the result random.
- **Threats against reliability:** These attacks neither wish to learn the voter's choice nor forge the result. Their main purpose is to make the election system break down or violate user's trust to the election system.

4.1 Threats Against Anonymity

1. **Side channel and subliminal channel attack**
   Since voters cast their votes through a voting machine, the voting machine will know the choice of the voter. An adversaries can use a corrupted voting machine, and some side channel or subliminal channel to learn this voter's choice, e.g. a corrupted voting machine can print ciphertexts such that the space between the first and second letters is larger than the space between the second and third letters when the plaintext is a vote for the Blue Party, otherwise Yellow Party. Hence by looking at the *voting ballot*, the adversaries can learn the choices of the voters.
2. **Kleptographic channel attack**
   The voting machine can carefully choose the random values so that the ciphertext can be read by colluding adversaries without decoding. An example, a corrupted voting machine can generate ciphertexts such that the seventh most significant bit is odd exactly when the plaintext is a vote for the Blue Party, otherwise Yellow Party.
3. **Authority knowledge attack**
   Even if the voting machine is honest in the previous two attacks, it has the power to retrieve voter's choice just by reading the *voting ballot*. The voting machine can implement this attack using two methods. One is to record all relationships between ciphertexts and their plaintexts when generating ballots. The other is to remember all relationships between the identifier r and the seed generator g of this ballot. Then by reading r on the *voting ballot*, the voting machine can find out the plaintext using the signature $sig_K(q, i, X, Z)$.
4. **Hidden camera attack**
   The authors have mentioned this attack in [7], that if the voting booth is monitored by hidden cameras, or if voters successfully bring cameras into the voting booth, voter privacy can be violated.

5. **Duplicating onion attack**
   Suppose now that an honest voter v uses a perfectly honest voting machine to cast her vote. A collusion between a corrupted voter v', the first mix server and the last mix server allows to learn the choice of v.
   Step 1: Before the mixing procedure, the corrupted voter v' casts her encrypted vote c_{aux} (four *Onions*) which is put on the bulletin board, and she sends a valid vote m_{aux} (four opened *Onions*, e.g. taken from her control ballot[1] or from any corrupted voting machine) to the last mix. Note that c_{aux} is not supposed to be an encryption of m_{aux}.
   Step 2: The first mix server removes c_{aux} and duplicates (with re-encryption) four *Onions* of voter v.
   Step 3: The last mix server opens all *Onions* and recognised the duplicated votes of v. Then she replaces one of the duplicated results by m_{aux}.
6. **Using ElGamal malleability – 1**
   If the first mix server colludes with the last server, they can find out how a voter v has voted as follows: the first mix server replaces all four received *Onions* $(c_i)_{1\leq i\leq 4}$ of v by ${c_i}^2 = ({m_i}^2 \cdot (y_1 \cdots y_\lambda)^{2k_1}, g^{2k_1})$. After decoding, the last mix server will obtain four messages which are not readable. Then m_i can be retrieved by calculating the square root of ${m_i}^2$. Finally, the last mix server correctly outputs all the votes, but she has learnt this voter's choice.
7. **Using ElGamal malleability – 2**
   This attack is much more powerful than the previous one. Let $(v_i)_{1\leq i\leq n}$ be the set of all voters. The first mix multiplies the four *Onions* $(c_{i,j})_{1\leq j\leq 4}$ of v_i by i (that is, multiply the first vote by 1, the second vote by 2, and so on). The last mix server decrypts all the votes and for any fixed $k \leq n$, she divides all the obtained results $(i \cdot m_{i,j})$ by k. She will recognise exactly the two votes of v_k (beginning by B or Y), e.g. $(B, r_U, ser_V, {sig'}_{K'}(B, r_U, 1))$ and $(B, r_U, ser_V, {sig'}_{K'}(B, r_U, 2))$, and the serial number ser_V allows the last mix server to recover the two identifiers $(r, ser_V, {sig'}_{K'}(r, 1, U))$ and $(r, ser_V, {sig'}_{K'}(r, 2, U))$. Hence the last mix correctly outputs all the votes, but she knows the choice of each voter.
8. **Suicide attack**
   Suppose that the first mix server is the only dishonest one. By using the methods of the previous three attacks, e.g. multiplying the *Onions* and dividing them after the final decryption step, the first mix server learns the choice of each voter. Since the decrypted votes are not valid ones, she will be caught thanks to the backtracking procedure, but it is too late.

4.2 Threats Against Correctness

1. **Voting machine colludes with the first mix server**
   A corrupted voting machine can always place the ciphertexts for the identifier after the ciphertexts for candidate (or it uses a subliminal channel to indicate the position of ciphertexts for the identifier), and the first mix replaces the first two *Onions* (ciphertexts for candidate) by *Onions* forged by the voting machine.

[1] The authors have not described any procedure to prevent reusing a control ballot.

2. **Voting machine colludes with the last mix server**
   The last mix server can replace some votes (but not identifiers) by valid votes forged by a corrupted voting machine.
3. **Voting machine colludes with any mix server**
   Since the voting machine has the ability to retrieve all identifiers just by reading the ciphertexts (before decoded by the first mix server) on the bulletin board, if it colludes with any mix server, it can generate a whole batch of forged votes (with the same identifiers). And the faulty mix server can use these forged votes to replace the original ones.
4. **Voting machine generate faulty ballots**
   Because the ballots are generated on demand and they are not distributed to voters randomly. Therefore, if some voters are coerced to cast their votes at some particular time (e.g. between 3pm and 4pm), and these voters are not allowed to audit their *control ballots*, the voting machine can forge ballots during this period without being detected. Another possibility of this attack is similar as introduced in [6], when voters cast their votes in the voting booth, the voting machine has to print the *hash ballot* before voters make their choices. Otherwise, the *hash ballot* may not contain commitment to the *voting ballot*. If some voters do not notice the difference, they may be provided with faulty ballots.
5. **Discarded receipt attack**
   When some voters are forced to surrender all their ballots, if the first mix server wish Blue Party to win, she can use the ciphertexts in the *control ballot* (for Blue Party and identifier) to replace the original *Onions*. Besides, if the voting machine colludes with the first mix server, they can forge valid votes to replace the discarded ones.
6. **Cast votes for absent voters**
   If some voters do not cast their votes, adversaries (especially some election authorities) may use their identity to cast votes.

4.3 Threats Against Reliability

1. **Early publishing**
   Since the voting machine has the ability to retrieve any voter's choice just by reading the bulletin board. It can reveal partial result which may affect voters before they cast their votes.
2. **Invalid signature**
   If the voting machine generate invalid signature on *control ballot* or *hash ballot* (since they will not be checked in the voting booth), it will be difficult to determine afterwards the invalid ballot is forged by the voting machine or by the faulty voter.
3. **Identifier collision**
   If the voting machine generate some ballots using the same identifier, the cheating will be detected at last. But voter's trust might be destroyed.
4. **Denial of service**
   The authors have not suggested that the private keys $(x_1, x_2, \cdots, x_\lambda)$ are threshold distributed among all mix servers. Hence the absence of at least one mix server will result in denial of service for the whole election system.

5 Conclusion and Discussion

We have shown a number of attacks to the scheme in [7]. Generally speaking, the reasons for these attacks include:

- The voting machine has too much power. It generate all ballots not in a distribute fashion and it has the ability to learn voter's choice. The security of this scheme is heavily relying on that the voting machine is honest.
- The mixnet suggested in the scheme is not fully verified. Therefore, some attacks can succeed without being detected. A suggested mitigation to some attacks is that the first and the last mix server have to be verified immediately after their decoding. Note that this cannot completely solve the problem, as the first and the last mix server may collude with the mix servers next to them. To ensure the correctness of the result, the mixnet needs to be fully verified by public, e.g. using techniques in [13].
- The authors have not described any procedure to prevent reusing the *control ballots*. Thus if they are improperly used, the final result will be inaccurate. We suggest that all opened *control ballots* should be published onto the bulletin board as well, and one has to ensure that no unit in the final result is duplicated from the opened *control ballots*.

References

1. Punchscan: http://www.punchscan.org
2. Adida, B., Rivest, R.: Scratch & Vote: Self-contained paper-based cryptographic voting. In: Proceedings of the 5th ACM workshop on Privacy in Electronic Society, pp. 29–40 (2006)
3. Chaum, D.: Secret ballot receipts: true voter-verifiable elections. IEEE: Security and Privacy Magazine 2(1), 38–47 (2004)
4. Chaum, D., Ryan, P., Schneider, S.: A practical voter-verifiable election scheme (LNCS 3679). In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 118–139. Springer, Heidelberg (2005)
5. Jakobsson, M., Juels, A., Rivest, R.L.: Making mix nets robust for electronic voting by randomized partial checking. In: Proceedings of the 11th USENIX Security Symposium, pp. 339–353 (2002)
6. Karlof, C., Sastry, N., Wagner, D.: Cryptographic voting protocols: A systems perspective. In: Sagiv, M. (ed.) ESOP 2005. LNCS, vol. 3444, pp. 186–200. Springer, Heidelberg (2005)
7. Klonowski, M., Kutylowski, M., Lauks, A., Zagorski, F.: A practical voting scheme with receipts. In: Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 490–497. Springer, Heidelberg (2005)
8. Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Proceedings of the 8th ACM conference on Computer and Communications Security (CSS 2001), pp. 116–125 (2001)
9. Neff, C.A.: Practical high certainly intent verification for encrypted votes. VoteHere document (2004), http://www.votehere.net/vhti/documentation
10. Neff, C.A.: Verifiable mixing (shuffling) of ElGamal pairs. VoteHere document (2004), http://www.votehere.net/vhti/documentation
11. Ryan, P., Peacock, T.: Prêt à Voter: A system perspective. Technical Report of University of Newcastle, CS-TR:929 (2005)

12. Ryan, P., Schneider, S.: Prêt à Voter with re-encryption mixes. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, Springer, Heidelberg (2006)
13. Sako, K., Kilian, J.: Receipt-free mix-type voting scheme. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 393–403. Springer, Heidelberg (1995)
14. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology, 161–174 (1991)

Secure Internet Voting with Code Sheets

Jörg Helbach[1] and Jörg Schwenk[2]

[1] Sprint Sanierung GmbH, D-51061 Köln
joerg.helbach@sprint.de
[2] Ruhr-University Bochum, D-44780 Bochum
joerg.schwenk@rub.de

Abstract. Malware on Personal Computers is a major security issue today. This fact implies that all solutions intended to secure Internet-based voting have to be re-evaluated under the assumption that a local malware application is capable of controlling the interface between user and PC. We propose to use paper-based code sheets, originally introduced by Chaum, to overcome this problem, and for the first time give a security analysis of this solution. We show that a modified, 3-step-scheme, can be considered secure against local malware attacks. Our scheme could then particulary be used to held shareholder elections or votes in an associaton over the Internet.

1 Introduction

Since 2004, malware has evolved from the playground of script kiddies to a valuable tool for cyber criminals. Trojan horses have been found in the wild by AV companies that control the complete input/output of browser based online-banking applications, even if there is an authentic SSL connection with the bank [Gri06][Emi06]. According to Vinton Cerf, co-developer of TCP/IP and architect of the Internet, the problem is immense: "Of the 600 million computers currently on the Internet, between 100 and 150 million were already part of these botnets" [Web07]. Against this background, it seems naïve to propose to perform elections over the Internet. However, political elections in Estonia were held over the Internet (fortunately only about 3% of all voters voted this way, which resulted in statements from officials that this kind of voting was secure) [Est07] and a new EU directive states that shareholders must be allowed to vote over the Internet [Zyp07]. We propose a solution introduced originally as "SureVote" by David Chaum [Cha01], and, up to our knowledge, for the first time analyze its security. In this scheme, the voting agency prints paper sheets as can be seen in table 1, which contain the list of voting alternatives (e.g. names of candidates or yes/no) together with two random numbers, which are chosen separately for each sheet and each voting alternative on this sheet. The first number, which we call Voting Transaction Number (or Voting TAN for short), must be entered into the web form, and a successful cast of the vote is communicated to the voter by displaying the second number, thus called Confirmation TAN. Our security model completely differs from "classical" security models for electronic voting

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 166–177, 2007.

in that we consider the voting authority to be trustworthy, and the personal computing devices of the voters (i.e. their PCs) to be insecure. Thus the voter has no possibility to perform cryptographic computations, because he simply can not trust the outcome of these computations.

2 Related Work

2.1 Voting Requirements

Generally, all elections have to be at least free, universal, secret and equal. In some states elections also must be direct. Many research has been done to adopt voting systems to this properties. Summarized, there are five requirements and threats a voting system has to deal with:

- **Universal election.** A first threat is to prevent the voters from voting. For that purpose an attacker can e.g. use a denial of service attack. Either the voting servers or clients are prevented to provide their services or the communication channel is made unavailable.
- **Direct election.** The second threat is that the ballot is modified before cast. E.g. a Trojan horse or a man in the middle could invalidate a ballot by adding too many additional votes.
- **Secret election.** Third, the loss of the voter's privacy is possible, i.e. the secrecy of the ballot is not guaranteed. As every election has to be free, universal, equal and secret, the basic principles of elections would be violated.
- **Equal election.** Another threat is that a voter could try to cast more than only one ballot.
- **Free election.** The last threat is vote buying, selling and trading. That is, if a voter sells his voting decision to an attacker. Therefore, a voter must not be able to prove his decision to the vote buyer.

An issue that all those threats have in common is scaling, because it is very easy to use computers for automated attacks. If a voting system is vulnerable to one or more of those threats, it mostly fails on a large scale.

The voting scheme based on code sheets deals with all those issues, except of DoS attacks. Vote selling is a problem that is difficult to solve with a purely paper-based solution, because a voter may simply sell the code sheet. A partial solution to this problem is to use vote updating to receive receipt-freeness. However, vote selling is tolerable or even allowed in some non-political elections, e.g. the election of a CEO by shareholders of a company.

2.2 Voting Technologies

In general, all proposed voting systems use different technologies to meet those requirements.

Homomorphous encryption. The first approach uses homomorphous encryption, so that $e(x_1) + e(x_2) = e(x_1 + x_2)$ where x_1, x_2 are ballots and $e(x)$ is an

encrypted ballot. A single ballot is never decrypted even when computing the tally. A voting system, which uses homomorphous encryption, is e.g. CyberVote[1] [Cyb03]. The voting authorities publish all encrypted ballots on a bulletin board. Additionally all ballots are added. Then just the result of the addition is decrypted, so everyone can prove that the tallying and the addition were correct. Using anonymous channels this voting scheme can receive receipt-freeness, universal verifiability, privacy and uncoercibility.

Mix-Nets. The second approach, using Mix-Nets, permutes the messages on the communication channel, so that the order of the incoming ballots is changed to guarantee the anonymity of the voters. Mix-Nets are often used in combination with homomorphous encryption systems.

Blind signatures. The third type of a voting system uses (blind) signatures. Here the voter sends his encrypted and signed ballot to a canvasser, who checks if the voter is eligible. In this case, the canvasser signs the ballot himself and sends it back to the voter. Now the voter can remove his own signature, so that he gets an encrypted ballot, which was signed by a canvasser. The voter can now send that ballot to the urn, which can decrypt and count it. This approach was firstly introduced 1982 by David Chaum [Cha82]. A voting system used in practice is e.g. Polyas[2] [Mic06]. Based on this concept, many election protocols have been proposed. A very important voting scheme based on blind signatures is the system Fujioka, Okamoto and Ohta proposed in 1992 [Fuj92]. In this scheme the counter publishes the encrypted ballots, so every voter can check if his ballot is on the list. The problem here is that this voting scheme is not receipt-free, i.e. the voter can prove his decision to a third-party and thus can sell his vote.

Paper based voting systems for general elections. Lately, some concerns have been raised about the acceptance of voting systems based on cryptographic functions that will not be understood by a majority of voters, or whose implementation cannot be verified publicly. As a consequence, we have seen a renaissance of paper based voting schemes, which cleverly combine simplicity and security. An important property of an election system is, whether the voter can verify, that his ballot had been computed correctly in the tally. In 2004 Chaum, Ryan and Schneider proposed a voter-verifiable voting scheme, Prêt-a-voter [Cha04]. In that voting scheme the ballot is paper-based and consists of two columns. On the left column the candidates are listed in a random order. On the right one the corresponding bubbles are printed, where the voter can make his choice. Additionally on the right column an encrypted code is printed, which contains the information how to reconstruct the order of the candidates, without having the left column. In an election process a voter gets two paper-ballots. The voter selects randomly one of those ballots. The voter checks this ballot, if the random

[1] Several elections were conducted with a prototype of the CyberVote voting system. All cast ballots were no real votes, but just test ballots. An improved variant of CyberVote is currently commercialized.

[2] Polyas has been used for the chairmanship elections in 2004, 2005 and 2006 of the German association of computer science. About 12.000 real votes were cast.

order of candidates on the left side suits the encrypted code on the right. This checks can be done in different ways, e.g. a dummy vote or giving the encrypted code to a canvasser, who returns the order of candidates[3]. If the ballot is correct, one can assume that the second ballot is correct, too, and the first paper ballot can be destroyed. On the second ballot the voter marks his choice, detaches the left column and destroys it. The remaining ballot is fed into an optical reader, which reads the position of the voters cross and the encrypted code.

In 2006 Adida and Rivest enhanced Prêt-a-voter [Adi06]. On the right column the encrypted code is exchanged by a 2D-barcode, which contains the order of candidates, and a scratch surface. The voter makes his choice as described above. After the left column has been detached and destroyed, the voter presents the remaining ballot to a canvasser, who checks and detaches the scratch surface. Then the ballot is scanned and published on a bulletin board, so the voter can check, if his ballot has been computed in the tally.

Another paper based voting system is Rivest's ThreeBallot Voting System [Riv06]. This system doesn't use any cryptography, but only paper ballots. A unique number identifies every ballot. Except this unique ID, all ballots are identical. The voter casts three ballots and gets a copy of one of his ballots as a receipt. The user can choose, which ballot he wants to copy. To vote for a candidate the voter must fill in two of the three bubbles for this candidate, all other candidates must have one bubble filled in, i.e. the voter has to fill in at least one bubble in each row and may not fill in all three bubbles in a row. With these requirements the tallying is quite simple. To receive the election result for a candidate, one just has to count all marked bubbles for this candidate and subtract the total number of voters. As all those voting schemes are paper-based, they cannot be easily adopted to a remote voting system.

2.3 Voting Systems for Internet Elections

Besides those improvements for paper ballots or direct-recording voting machines, which are running in a voting booth inside a polling station, researchers also developed voting systems, which are used completely over the Internet. In the literature these systems are often named Remote Online Voting Systems. Early remote online voting systems were Sensus by Lorrie Cranor [Cra97] and E-VOX by Mark Herschberg [Her97]. Both Sensus and E-VOX are based on the Fujioka, Okamoto, Ohta protocol, mentioned above. Obviously, remote online voting systems have to deal with the same requirements as voting systems in general. Additionally those systems have to ensure the voters' authentication over the Internet, i.e. that the voter is eligible and has not voted before. It seems to be easy to authenticate a voter, but, however, as the election has to be anonymous, too, it is one of the main technical problems to bring authentication and anonymity together. There are different possibilities to authenticate a voter. Authentication can be done by issuing a username and a password to all voters, with what the voters can login into the voting application. Another

[3] For details see [Cha04].

possibility is to authenticate by SmartCard. Certainly, it must be guaranteed, that all polling stations, i.e. the personal computer, are attached to an adequate card reader. Biometric authentication is also possible, but research into biometric authentication showed, that the false rejection rate is currently too high for such a sensitive environment as electronic voting [Hof04]. It is obvious that the voting systems, which use one of those authentication methods, have to deal with the anonymity of the voters. A fourth alternative is the usage of a transaction number. Each voter is issued such a number at random, so anonymity of the voter is given. Other problems using remote online voting systems are the "Secure Platform Problem", introduced by Rivest [Riv02], and denial of service attacks on the voting servers. In 2001 Chaum proposed an approach to deal with malware on the client computers, using code voting [Cha01]. In 2002 Oppliger referred to code voting as code sheets [Opp02]. On every ballot for each candidate is generated a random number. The voter makes his choice by submitting that random number to the voting server. The server responds with a verification number. That voting scheme is not receipt-free, for the voter can prove his vote by giving his code sheet and the verification number to a coercer. To receive receipt-freeness in some voting systems is used vote updating [Vol06][Ram02]. A voter can overwrite his decision as many times as he wants. Only the last vote is computed in the tally.

3 Trust Model

Traditional democratic elections are protected by the direct involvement of many people from different political parties in the organization of the election[4]. This guarantees that no single person or no small group of persons can modify the outcome of the election, as it is often the case in non-democratic elections. However, we must stress the fact that only a relatively small subset of all voters is involved here. A main concern of modern voting schemes is to broaden the group of people that can check the correctness of the voting process [Sch00]. However, regarding remote online voting systems this comes at the cost of performing highly complicated cryptographic computations by each voter. The only tool for this task that is available today is the personal PC of the voter. The main drawback of this highly democratic approach is that today many of the PCs are no longer under the control of the voter. To give just one example: In May 2006, an American botmaster who controlled about 400.000 personal computers was sentenced to 57 months in prison [Reg06]. This single person could therefore have controlled nearly half a million votes. Our trust model is therefore based on the assumption that the voter's PC is untrusted. We assume the existence of a trusted voting centre that issues printed code sheets and which is controlled by a democratic selection of people from all political groups. That is, all printed code sheets are correct and distributed to all eligible voters. The voting servers and databases are reliable, secure and trustworthy. We further assume, that the tallying is trustworthy, i.e. the voting authority doesn't add or delete any valid

[4] Separation of duties (SoD).

Table 1. Printed code sheet

Voting TAN	Candidate	Confirmation TAN
738747987	Ronald Reagan	332676873
983293774	Bill Clinton	676476488
192851911	Will Smith	301287123
...	...	...

votes. Another assumption is that vote selling and buying is either not permitted or even allowed as it is in some association elections or elections for shareholders [Zyp07].

4 Voting Scheme

In our voting scheme, we assume that distributing the code sheets by snail mail is relatively secure; i.e. that attacks are possible, but that the percentage of attacked code sheets is negligible compared to the total number of votes. In other words: attacks do not scale! In contrast to this, we do not make any assumptions regarding the security of either the communication channel through the Internet, or on the security of the client PC used for voting. Nevertheless we are able to prove that no attacker is able to perform successful attacks other than DoS.

The voting scheme works as follows:

Phase 1: Setup. In the Setup phase, the trusted voting authority generates the voting TAN lists. For each list and for each candidate, a random voting TAN and a random confirmation TAN is chosen. This data is printed on a code sheet, and stored as a record in the secure database of the voting centre. The addresses of voters are printed on envelopes, and the TAN lists are inserted into the envelopes in a random order. Then the lists are sent by snail mail to the voters. This process can be democratically assured by controlling the permutation of code sheets (e.g. by manually shuffling the lists) and by randomly selecting samples from the outgoing mail to check whether it contains a valid TAN list. (An additional randomization effect could be achieved by exchanging code sheets at "voting parties". This would improve the democratic trust in code sheets. However, it must be guaranteed that the authenticity of code sheets can be easily verified.)

Phase 2: Voting. Each voter opens the voting web page and enters the voting TAN printed left of the chosen candidate's name. As a result, the voting web page displays the confirmation TAN so that the voter can check that he did not make any typing errors. To provide receipt-freeness, even though the voter has a confirmation TAN and his TAN list to prove his decision, vote updating[5] is allowed. That is, multiple casts per VoterID are possible, but only the last vote counts[6]. If the chosen voting TAN is not accepted or the voter gets a

[5] See also [Ram02].

[6] A detailed analysis of multiple casts is given in [Vol06].

false confirmation TAN, he or she should claim to the voting authority. As we assume, that the voting authority is trustworthy and the distributed code sheets are correct, one can conclude that in this case either the voter made a mistake or the voting client is infected.

Vote Updating. Vote updating can be used to minimize the threats from DoS attacks and vote buying. If vote updating is allowed, the voter can be encouraged to vote early (because he can modify his decision later), and thus effects of DoS attacks in the final stage of an election can be minimized. On the other hand, if a voter sells his code sheet, he may keep a copy and try to update his vote very lately. However, even if the vote-seller tries to update his or her vote, he or she is racing with the vote-buyer, who can arrange to be almost certain to win the race, since he can re-perform the update as many as times as needed. We have to assume that the vote-buyer probably has more resources and patience than the vote-seller, and for instance can automate the process of repeatedly sending updates.

Phase 3 (optional): Public verification of the election result. The result of the election can be made publicly verifiable by publishing a list of all cast voting TANs (last vote in case of vote updating) together with the candidate they represent, sorted by voting TAN. Each voter can then check that his vote was counted correctly, and compute the number of votes for each candidate. The drawback of this solution is that it facilitates vote selling because the voter is able to prove what he voted by showing the code sheet to the buyer. However, we assume a scenario, where vote selling is either not permitted or even allowed.

5 Security Analysis

In this section we will show, that our proposed voting scheme is secure and only allows denial of service attacks and vote selling. As mentioned before, there are several threats a voting system has to deal with. For all those threats we analyze, which type of attack is remarkable and if our voting scheme is vulnerable to those attacks.

5.1 Communication Model

In our model, the human voter directly communicates with the voting server. To protect this communication against active and passive adversaries, we use ideal cryptographic functionality. In this model, integrity and confidentiality of voting ballots are not protected by encryption and/or digital signatures, but by sending a valid handle to a database entry:

- When the Voting TAN is sent over the Internet, the confidentiality of the candidate's name is protected by the fact that the TAN is only a randomly chosen handle to this name, known only to the central database and to the voter (via the code sheet).

– The integrity of the Voting TAN is protected by the fact that the TANs are sparsely distributed in a large number space. Any attempt to fake a Voting TAN will thus with high probability result in a number not belonging to this sparse set.

5.2 Attacks by a Passive Adversary

A passive adversary is only able to attack the secrecy of an election. In our model, the passive adversary can observe the Voting TAN entered into the web browser, and the confirmation TAN sent by the voting server. Since he by assumption has no access to neither Code Sheet nor to the database of the voting server, and since both TAN numbers were chosen at random, he can not do better than simply guess the vote. This guess may be biased by the result of the election, which does not give any additional information about the individual vote.

5.3 Attacks by an Active Adversary

In addition to the abilities of the passive adversary, an active adversary may insert any message into the communication channel between the human voter and the voting server. This includes sending TAN numbers, which have been observed or randomly chosen, in both directions, and error messages of any kind. Error messages are a means to trigger some attacks based on social engineering, as explained later.

Disenfranchisement/DoS. In our context disenfranchisement is preventing the voter from participating in the election. For this, an attacker can perform a denial of service attack (DoS). In our context a DoS is the attempt to make the voting system unavailable to the voters. Our proposed voting scheme cannot deal with general DoS. However, we make the assumption that DoS attacks based on local malware can only prevent a fixed (small) fraction of voters from voting. The advantage of our solution is that denial of service attacks could be detected.

Modifying a ballot: Voting for a specific candidate. In this attack the adversary tries to send Voting TANs to the server that are associated with a specific candidate. To this purpose, he can guess or steal Voting TANs.

– If he guesses a Voting TAN, he can not vote for a specific candidate, since the valid Voting TANs are equally distributed over all candidates.
– If he steals a Voting TAN, he gets no information about which candidate this TAN is associated with, since the Voting TAN was chosen randomly. This voting TAN can only be used for the candidate chosen by the voter.

Modifying a ballot: Averaging attack. In this attack the adversary simply guesses a Voting TAN, and sends it to the voting server. If his guess was correct, the voting server answers with a Confirmation TAN. Since valid TANs are equally distributed over all candidates, this attack will have an averaging effect amongst all candidates: favorite candidates will get less votes, outsider candidates will

get more votes. Since vote updating is allowed, this attack will proceed in two stages: (1) In the first stage, which starts at the beginning of the voting period, the adversary guesses Voting TANs, and stores successful guesses in a database. (2) Towards the end of the voting period, all valid Voting TANs stored in the database will be cast again, to overwrite the votes of legitimate users. To make the voting system resistant against this kind of attacks, the averaging effect must be minimized such that it has no effect on the voting outcome. The following parameters are important here:

- The number n of guessed Voting TANs the adversary is able to transmit to the voting server during the voting period. This number depends on several factors: the number of PCs controlled by the adversary, the number of invalid votes accepted per PC/browser, ...
- The number m of digits of the Voting TAN.
- The number k of candidates.
- The number v of voters.

To guarantee that this attack has no effect on the outcome of the election, the number α of votes per candidate that can be guessed must be negligible.

$$\alpha = \frac{n \cdot v \cdot k}{k \cdot 10^m} = \frac{n \cdot v}{10^m}$$

E.g. if $n = 10^6$ and $v = 10^8$, then choosing $m = 15$ would guarantee that less than one vote per candidate can be generated this way.

Modifying a ballot: Vote-for-the-outsider attack. In this attack, the adversary substitutes the Confirmation TAN repeatedly by an error message. The voter is thus tricked into believing that there is a printing error in the Voting TAN of his favorite candidate. A possible reaction is to test another voting TAN (since vote updating is allowed). It may be argued that the voter will not use the TAN of the biggest competitor of his candidate, but an outsider. This is a very severe attack, since its impact only depends on the number of PCs controlled by the adversary, and cannot be decreased by e.g. increasing the length of the Voting TAN. The attack can be triggered by other events, e.g. a candidate changing his political direction from left to right (or vice versa), to win new votes, and inject old votes recorded by malware.

Table 2. Printed code sheet with Finalization TAN

Voting TAN	Candidate	Confirmation TAN	Finalization TAN
738747987	Ronald Reagan	332676873	442367810
983293774	Bill Clinton	676476488	123456789
192851911	Will Smith	301287123	520172861
...	...	...	...

A solution to this problem is depicted in table 2. Each vote consists of two parts: the voting TAN and the finalization TAN. The vote will only be counted if both parts arrive at the server. If we assume that no user enters the finalization TAN if he hasn't seen the confirmation TAN before, we can prove the security of our scheme. The problem we are facing resembles the classical (unsolvable) "two army problem" from communication theory [AEH75][Gra78]: Two parties (here the voter and the voting server) try to communicate through a channel controlled by the adversary. They can never reach a state where they have the same information, because if a message arrives at the receiver, the sender does not know if the message was successfully transmitted. Our solution is along the lines of three-way-handshakes used in computer communications (e.g. TCP handshake).

We assume that a voter never would enter the finalization TAN, if the voting server diesn't respond with the correct confirmation TAN according to the submitted voting TAN. Therefore, in the counting phase a ballot could have two possible patterns: First, both the voting TAN and the finalization TAN were submitted correct to the voting server. In this case the vote is valid, and hence is considered in the tally. This vote only could be overwritten by a valid voting TAN, finalization TAN combination, which belongs to the same code sheet. Second, the voting TAN was submitted correct to the voting server, but the finalization TAN wasn't submitted. In this case the voting client could have faced an attempt to defraud or an DoS attack. Those votes are not computed in the tally, but published on a special bulletin board. The voters are challenged to check, if his or her voting TAN is published on that particular board and hence should recast their vote using another voting client.

Assuming that the vote for the outsider attack is successful, one can identify different voting patterns at the voting server regarding the last submitted voting and finalization TANs. These patterns could be:

- V-TAN-outsider, V-TAN-favorite, V-TAN-outsider + F-TAN-outsider
- V-TAN-outsider + F-TAN-outsider, V-TAN-favorite, V-TAN-outsider + F-TAN-outsider
- V-TAN-outsider + F-TAN-outsider, V-TAN-favorite + F-TAN-favorite, V-TAN-outsider + F-TAN-outsider

Votes with those patterns are also published on the bulletin board, with challenging the voters to check their vote.

5.4 Vote Selling, Buying and Trading

Vote selling is a major problem in political elections. In our voting scheme we assume an elections where vote selling is legal, e.g. votes for a CEO of a stock company. In this case shareholders may legally trade their voting rights, so there is no reason to prevent this in a code based scheme.

However, if vote selling is prohibited, and because Internet voting can make it easy to buy a large scale of votes by automating the process, it is a demanding

task for a voting scheme, to eliminate the possibility of vote buying. In our proposed voting scheme a voter has different possibilities to sell his decision:

- He can sell his code sheet to an attacker. This process is not scalable, because the code sheets are sent by snail mail to the voters. (This is comparable to sell a vote that is cast today in political elections by snail mail.) Furthermore, as the buyer cannot be sure that the voter didn't make a copy of the TAN list to vote at a later moment using vote updating, this possibility is very unattractive, because it results in a race condition.
- If public verification is used, a proof for correct vote selling may consist of a voting TAN sent to the buyer before the results of the election are published on the public bulletin board.
- As a possible countermeasure one could propose to include for each candidate A a fixed number i of invalid pairs (voting TAN, A) in the election result list, and to print on the code sheet one randomly selected pair for each candidate. However, in this case the buyer could simply request two voting TANs!

To summarize: Without public verification, code sheet based voting schemes are as vulnerable to vote selling as snail mail voting is today. With public verification, vote selling is as easy as sending a voting TAN to the buyer!

6 Conclusion

We proposed a very simple and small voting scheme, which is not vulnerable to most of the threats introduced by malware. This is achieved by using a 3-step-process where random TANs are exchanged between voter and voting server. An adversary can influence the absolute number of cast votes, but not their relative distribution.

Legal problems are another issue. How the case of even small DoS attacks can be handled legally is outside the scope of this paper, and vote selling may become a problem if code sheets are used on large scale.

References

[AEH75] Akkoyunlu, E.A., Ekanadham, K., Huber, R.V.: Some Constraints and Tradeoffs in the Design of Network Communications. ACM SIGOPS Operating Systems Review 9(5), 67–74 (1975)

[Adi06] Adida, B., Rivest, R.: Scratch & Vote. In: Proceedings of the 5^{th} workshop on privacy in the electronic society, pp. 29–40 (2006)

[Cha82] Chaum, D.: Blind signatures for untraceable payments. In: Advances in cryptology, pp. 199–203 (1982)

[Cha01] Chaum, D.: Sure Vote: Technical Overview. In: Proceedings of the workshop on trustworthy elections (WOTE 2001), presentation slides (2001), http://www.vote.caltech.edu/wote01/pdfs/surevote.pdf

[Cha04] Chaum, D., Ryan, P., Schneider, S.: A Practical, voter-verifiable Election Scheme, Technical Report 880, School of computing science, Newcastle university (2004)

[Cra97] Cranor, L.: Sensus: A Security-Conscious Electronic Polling System for the Internet. In: Cranor, L. (ed.) Proceedings of the Hawai'i International Conference on System Sciences (1997)

[Cyb03] Website of the CyberVote Project, http://www.eucybervote.org

[Emi06] Emigh, A.: The Crimeware Landscape: Malware, Phishing, Identity Theft and Beyond. A Joint Report of the US Department of Homeland Security–SRI International Identity Theft Technology Council and the Anti-Phishing Working Group

[Est07] Estonia to hold world's first Internet election, http://www.pcpro.co.uk/news/105714/estonia-to-hold-worlds-first-internet-election.html

[Fuj92] Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, Springer, Heidelberg (1993)

[Gra78] Gray, J.: Operating Systems. LNCS, vol. 60, pp. 393–481. Springer, Heidelberg (1978)

[Gri06] Grimes, R.: An SSL trojan unmasked, http://www.infoworld.com/article/06/03/03/75970_10OPsecadvise_1.html

[Her97] Herschberg, M.: Secure Electronic Voting Over the World Wide Web, Master Thesis, at the Massachusetts Institute of Technology (1997), http://theory.lcs.mit.edu/~cis/voting/herschberg-thesis/index.html

[Hof04] Hof, S.: E-Voting an Biometric Systems? In: Proceedings Electronic Voting in Europe Technology, Law, Politics and Society. LNI P-47, pp. 63–72 (2004)

[Mic06] Polyas Website, Micromata Objects GmbH, http://www.polyas.de

[Opp02] Oppliger, R.: How to Address the Secure Platform Problem for Remote Internet Voting. In: Proceedings 5^th Conf. Security in Information Systems (SIS 2002), vdf Hochschulverlag, pp. 153–173 (2002), http://www.ifi.unizh.ch/~oppliger/Docs/sis_2002.pdf

[Ram02] Wu, C., Sankaranarayana, R.: Internet Voting: Concerns and solutions. In: International Symposium on Cyber Worlds: Theories and Practices (2002)

[Reg06] http://www.theregister.co.uk/2006/05/09/botnet_master_ancheta_jailed/

[Riv02] Rivest, R.: Electronic voting. In: Syverson, P.F. (ed.) FC 2001. LNCS, vol. 2339, pp. 243–268. Springer, Heidelberg (2002)

[Riv06] Rivest, R.: The ThreeBallot Voting System. unpublished draft, version 10/1/06, comments appreciated, http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf

[Sch00] Scheier, B.: Voting and Technology. Crypto-Gram (2000), http://www.schneier.com/crypto-gram-0012.html

[Vol06] Volkamer, M., Grimm, R.: Multiple Casts in Online Voting: Analyzing Chances. In: Electronic Voting (2006)

[Web07] Weber, T.: Criminals 'may overwhelm the web, http://news.bbc.co.uk/1/hi/business/6298641.stm

[Zyp07] Zypries: Virtual general meetings. throughout Europe enhance the rights of shareholders, http://www.bmj.de/files/-/1739/Press%20Release%20in%20english.pdf

CodeVoting
Protection Against Automatic Vote Manipulation in an Uncontrolled Environment

Rui Joaquim and Carlos Ribeiro

ISEL / INESC-ID
rjoaquim@cc.isel.ipl.pt,
carlos.ribeiro@tagus.ist.utl.pt

Abstract. One of the major problems that prevent the widespread of Internet voting is the vulnerability of the voter's computer. A computer connected to the Internet is exposed to virus, worms, spyware, malware and other threats that can endanger the election's integrity. For instance, it is possible to write a virus that changes the voter's vote to one predetermined vote on election's day. It is possible to write such a virus so that the voter would not notice anything wrong with the voting application. This attack is very dangerous because it may pass undetected. To prevent such attack it is necessary to prevent automatic vote manipulation at voter's computer. Here we present CodeVoting, a technique to create a secure communication channel to a smart card that prevents vote manipulation by the voter's PC, while at the same time allows the use of any cryptographic voting protocol to protect the election's integrity at the server side of the voting application.

Keywords: Internet voting, vote manipulation.

1 Introduction

Remote electronic voting can be a powerful tool for our democracies. It can allow citizens to vote from anywhere at anytime and also provides faster vote count. However, it also brings some risks. Risks to the integrity of the election such as the risk of automatic vote manipulation, and risks to the privacy of voters that could ruin one of the pillars of our democracies. These risks apply both to the client and server side of the voting application.

Currently, cryptographic mechanisms can be employed to provide protection at server side. Techniques, such as digital signatures and zero knowledge proofs are normally used to guarantee correct vote handling. To protect voters' privacy other cryptographic techniques are used such as homomorphic ciphers, mix-nets and blind signatures.

Cryptographic voting protocols work at server side based on the assumption that several parties do not collude. On the other hand we have the client side of the voting application. The client application is centralized, i.e. not divided over several opposite entities, therefore it is usually considered "trusted". The

A. Alkassar and M. Volkamer (Eds.): VOTE-ID 2007, LNCS 4896, pp. 178–188, 2007.

client application shows the ballot, collects the answer, performs the voting protocol and verifies it on behalf of the voter. However, there is a problem with this approach, and the problem is considering the client application "trusted". The client application can be manipulated to cast a vote on a predetermined candidate while it misleads the voter into believing that her vote is the one that is cast. Another week point at client's side is the possibility of compromising the client application, or the remote computer where the client program executes, to facilitate vote buying and coercion. If the voter uses a computer that is controlled by the vote buyer or coercer it is easy to produce a vote receipt.

There is much work on the vote buying/coercion problem [4,5,9,10,11,6,12,15] under the assumption of a secure voter's platform. However, there is few on the insecurity of voter's platform. The work on vote buying and coercion free voting systems follows two main approaches. The first one is the use of a secure and secret communication channel between the voter and a trusted party of the voting system, allowing the voter to cheat on the buyer/coercer. Depending on the voting system, such secure and secret communication channel can be required prior to or on the election day. The second main technique used to prevent vote buying/coercion is allowing the voter to vote several times. Therefore, since usually the list of voters who voted is public, the only real vote buying/coercion possible that gives the attacker 100% of guarantees is to buy/coerce the voter to abstain.

We understand that vote buying/coercion is a potential big problem on Internet voting systems. However, we consider that the use of an insecure voters' platform can have a potential higher risk to election's integrity. We base our opinion on three reasons.

- First, large scale vote buying/coercion, involving possibly thousands of voters, is quite unlikely to pass undetected.
- Second, with all the security flaws on operating systems and applications, it is easy to write a virus that would be active on election's day to change the voter's vote.
- Third, we believe that writing a virus and disseminating it would be cheaper and more difficult to trace back to the authors than a vote buying/coercion attempt of a thousand voters, therefore more appealing to an attacker.

Therefore, we can say that the client side weaknesses of a remote electronic voting system is the main issue that prevents the widespread of remote electronic voting [2,7,8,13].

CodeVoting addresses some problems of the insecure voter's platform, namely automatic vote manipulation. CodeVoting works by creating a secure communication channel to a trusted component of the voting system. Therefore, CodeVoting can be considered as a kind of user interface to the voting system.

We propose the use of a tamper resistant device, such as a smart card, to be the trusted component of the voting system at client's side. Since a smart card provides a much more secure execution platform than an of-the-shelf PC, using CodeVoting to create a secure communication channel to the smart card,

throughout the voter's PC, will prevent automatic vote manipulation by malicious software installed in the voter's PC. Nevertheless, the voter's PC network and I/O capabilities will still be used to interact with the voter and the server side of the voting system.

The use of secure devices, e.g. smart cards, in electronic voting is not new [5,11]. Secure devices are typically used as a way to provide secure voter's authentication and for the generation and secure storage of secret values of the voting protocol. However, there is always the need to use the voter's computer to show the ballot and collect the answer, and usually this is assumed to be performed by a trusted vote client application. Our goal it to show how one can build a simple, secure and private communication channel between the common voter and her smart card, without the need to trust on the voter's computer. Secure communications channels are easy to achieve between machines, e.g. by sharing a secret key. However, making a secure and private communication channel between a machine a common human being is not so straightforward. The challenge is to keep the complexity of the communication channel as small as possible so the human can deal with it.

1.1 Vote Manipulation Attacks

A vote manipulation attack can be a modification of the voter's vote. The vote modification attack can be performed in two ways: i) changing the vote to a predetermined candidate or ii) changing the vote to a random candidate. While the first attack is more powerful the second may be easier to prepare in advance. By other words, to change a vote to a predetermined candidate one must have the knowledge of which candidate one wants to change the vote to, while to change the vote to a random candidate there is no need to know the candidates in advance.

If one wants to boost the number of voter for candidate A it is preferable to perform an attack to directly change the votes to votes for candidate A. On the other hand, if one wants the decrease the votes for candidate B, it suffices to perform a random vote modification attack in an area known to be much favourable to candidate B.

The other kind of vote manipulation attack is to fake a successful vote delivery. In many voting systems this can be done just by presenting the message "Your vote was successfully delivered. Thank you for voting.". This attack allows an attacker to reduce the votes on a candidate just by targeting an area with great affinity for that candidate.

In the next section we describe current proposals to minimize the weaknesses at the client side. In Sect. 3 we present Code Voting, a solution that prevents vote manipulation at client side. Then, we present in more detail the main components of CodeVoting in Sect. 4 and 5. We evaluate CodeVoting resistance to vote manipulation attacks on Sect. 6. Finally, we present some issues and future work on CodeVoting in Sect. 7 and conclude in Sect. 8.

2 Related Work

Cryptographic voting protocols can prevent vote manipulation at server side but that's only relevant if those properties cannot be easily broken at the uncontrolled client side of the remote voting system. Here we present an overview of the current proposals to deal with this problem.

One proposal is to restrict remote electronic voting to controlled environments [2], such as controlled voting kiosks, providing immediate protection on the common threats of uncontrolled environments, such as virus and other malicious programs. In a controlled environment it is also possible for election officials to verify that the correct client application is installed and running. However, this solution as the disadvantage of restricting voter's mobility and does not really work in the presence of corrupted voting officials who can install a malicious vote client application.

In 2001, Chaum [3] presented SureVote. SureVote allows the voter to vote using secret vote and reply codes. These secret codes allow the voter to detect if anyone changed the vote code during the voting process. SureVote consists in the generation of a secret vote and reply code for each candidate and for each voter. The codes are delivered to the voters prior to the election day. On election day the voter sends the vote code of her favourite candidate through the voting channel, e.g. Internet. At server side the reply code is computed by a set of trustees and sent to the voter that confirms it to verify that there was no vote modification. After the election end the trustees compute the real votes from the vote codes and publish the results. However, if there is at least one corrupted trustee, SureVote does not guarantee that in the counting phase the vote code is translated to the right candidate. Another issue with SureVote is that to protect against vote tracking and personification it is necessary to secretly create and anonymously deliver the vote codes to the voters, a requirement that is very hard to achieve and that creates new opportunities to attack the voting system.

A SureVote similar voting scheme was also used in the UK to enable the use of Internet, SMS and telephone voting channels [16]. The drawbacks of this system are the same than the ones of SureVote, i.e. it is necessary to guarantee that the codes are secretly generated and anonymously delivered to the voters, and there is no guarantees that the code vote is translated to the right candidate. The reply code only confirms that the vote has reached an entity that knows the right reply code.

Another proposal to solve the insecure platform problem is to use trusted computing technology [17]. Trusted computing is a technology that allows remote attestation of machines and programs running on them. With remote attestation it is possible to certify that the voter is using the correct voting program. Trusted computing also provides ways to secure I/O operations between the program and the physical I/O devices, therefore creating a secure environment for an application to run. The attestation process is based on measures performed on the software by a hardware module called trusted platform module (TPM). The client of a remote voting application needs to interact with the voter (I/O device drivers), needs to establish a connection with the voting server (network

protocol stack + network adapter driver) and, last but not least, it needs an environment to run on, i.e. what it really needs is a working operating system. The attestation of the core of the operating system, the device drives and the voting application can be cumbersome. Moreover, there are also problems concerning the maturity of the currently deployed technology [14] and concerning the revocation of cracked machines [1]. We believe that, for now, the application of trusted computing to remote voting as the only guaranty of correct application behaviour is not a valid alternative.

Kutylowski and Zagórski [10] propose the use of cryptographically hidden ballots. The voter requests n encrypted ballots that are randomized. Then she chooses one to use and verifies the other ballots with the help of a private external channel. If the revealed ballots are correct the voter uses the chosen ballot to cast her vote. The main disadvantages of this solution are the need of a private external channel to verify the ballots and the complexity of the voting protocol that requires the voter to deal directly with cipher texts.

The last proposal we are aware of replaces the standard PC's I/O by a secure I/O device [18]. The disadvantages of this solution are the necessity of a non standard I/O device, the corresponding costs, and the reduced I/O capabilities of the device used. However, it may be a possibility in a semi-controlled environment as a solution that allows adapting a standard PC to Internet voting.

3 CodeVoting

Reading the last section the reader quickly understands that none of the presented proposals is perfect. Here we present a proposal that does not restrict the voter's mobility, protects the vote even if the vote client application is executed in a malicious machine, and allows for the use of cryptographic voting protocols that protect voter's privacy and vote integrity at server side. Nevertheless, our system is simple enough to be used by the common voter.

Briefly, our solution consists in the following steps: i) the voter expresses her vote as a secret code, ii) the secret code is translated into the corresponding candidate code (clear vote), iii) the clear vote is used in a cryptographic voting protocol.

We propose the use of a tamper resistant device, such as a smart card, to be the trusted component of the voting system at client side. This trusted component we will call from now on VoterCard. The VoterCard will be used to securely authenticate the voter to the vote server, e.g. by means of digital signature, and it will also be in charge of i) the translation of the secret code to the candidate code, hiding the clear vote from the vote client application, and ii) the use of the cryptographic mechanisms provided by the cryptographic voting protocol. Additionally, it also provides proof of correct conclusion of the voting protocol.

3.1 CodeVoting Details

CodeVoting can be seen as a rearrangement of the ideas presented by Chaum [3]. However, the idea of CodeVoting is to use the codes just as a user interface and

not as the entire voting protocol. The secret codes are the base for the secure and private communication channel between the voter and her VoterCard. The voter will use secret codes to choose her favorite candidate. Each VoterCard has a set of secret codes associated with it that are printed on a paper card, the CodeCard. The details on how the voter gets her VoterCard and CodeCard are explained later.

For the voter the voting process is quite simple. The voter just uses a CodeCard to translate the candidate code into a vote code.

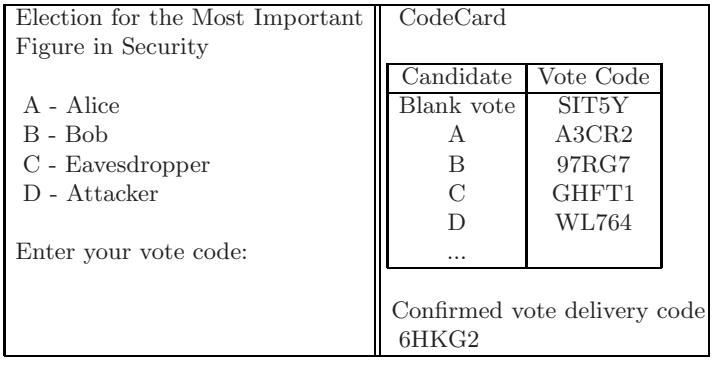

Fig. 1. Example of a ballot (on the left) and a CodeCard (on the right)

For example, a voter, with the ballot and CodeCard of Fig. 1, that wishes to vote for candidate D just have to enter WL764 as the vote code.

Every voter will have a different CodeCard, therefore different vote codes for the same candidate. Each CodeCard is associated to a VoterCard, which is responsible for the translation of the vote code to the candidate code. Only the voter and the VoterCard should know the codes written on the CodeCard. Therefore, CodeVoting protects against a malicious voting application trying to change the voter's vote.

After translating the vote code to the candidate code any voting protocol can be used to cast the vote. One solution is using the VoterCard to process the candidate code accordingly to the cryptographic voting protocol and submit the vote. Another alternative is to create an unchangeable vote and pass it to the voting client application running on the voter's PC so it can proceed with the voting protocol.

When the VoterCard receives a confirmation of a successful vote delivery it releases the confirmed vote delivery code, assuring the voter that her vote was successfully delivered.

Based on this overview of CodeVoting the reader can easily understand that CodeVoting is a kind of an user interface plugin to a voting system that can protect the voter's choice from manipulation.

4 VoterCard

As explained before, the VoterCard is in charge of the translation of the vote code to the clear vote, and also of the execution of a cryptographic voting protocol. However, we did not explained how each voter gets a VoterCard.

We propose to do the distribution of the VoterCards to the voters in a registration phase. This procedure is only required once, i.e. the VoterCard will be reused in subsequent elections. Since the VoterCard is a smart card we propose the use of it also as a secure voter's authentication mechanism, by means of digital signature. Therefore, a public key infrastructure (PKI) should be in place before the registration process. The PKI used can be set up just for elections' proposes or can be of more wide-use in a national e-Government project. This last approach can be useful to prevent at some level vote buying and coercion, because if the voter gives her VoteCard to a vote buyer/coercer it is not just a vote that the voter gives away, it is also all the e-Government rights of the voter.

4.1 Is the VoterCard Trustworthy?

Reading the description of CodeVoting, the reader quickly understands that CodeVoting relies on the correct behaviour of the VoterCard. Therefore one can ask: CodeVoting is designed to protect the voter from the insecure voter's PC, but what guarantees are given that the VoterCard does in fact do what it is supposed to do? One way to verify that the VoterCard does in fact what it is supposed to do is to test it. Besides testing the VoterCards in the production phase we believe that would be good to have additional random testing by an independent certification authority.

Additionally we can make voters also part of the certification process. It could be possible for a voter to verify her VoterCard by running a fake election with instant results sometime before the real election.

Nevertheless, one can point out that the application running inside the VoterCard is somehow able to detect that is being subject to a test, and therefore it will act properly in the tests but it will still change the voter's vote on the real election day. To prevent this scenario one must be sure of the software running inside the VoterCard. Fortunately, smart cards support signed applications. Therefore, and because it is possible to know which software is inside the VoterCard by verifying its signature, it is possible to use open source certified software. Of course, we can also have certified applications running on a PC. However, since it is possible for an attacker to take control of the voter's PC, a signed application does not guarantee correct behaviour. On the other hand, it is not possible to take control over the smart card. Therefore, an open source signed application can guarantee correct behaviour.

5 CodeCard

The CodeCard is just a paper card with codes printed on it. There should be one CodeCard per VoterCard so that every voter votes with different codes.

The voter should be the only one with access to the codes of her CodeCard to prevent manipulation of her vote. Consequently we have a problem to solve: how to create the CodeCard, associate it with the VoterCard and give it to the voter without leaking the codes.

We propose to generate each voter's CodeCard within the VoterCard. This is a good option because the CodeCard becomes automatically associated with the correspondent VoterCard and no other entity besides the VoterCard has access to the codes on the CodeCard. However, we still have the problem of how to secretly print the CodeCard, i.e. how to give it to the voter without leaking the codes. We think the best idea is to have a certified CodeCard printing machine, the CodeCard generator interface (CCGI), available at the local authorities' offices. Since the codes are generated inside the VoterCard the CCGI would be very simple. It would consist only of a smart card reader, a keypad (for inserting the PIN of the VoterCard and unlock it) and a small printer. We believe that such a simple hardware could be easily certified and sealed to ensure the secrecy of the codes printed. With the CCGI certified and in place a voter could go to any local authority office to generate a CodeCard for her VoterCard. For privacy reasons the CCGI should be inside a private booth, similar to the ones used for traditional paper based voting.

6 Evaluation

We defend that CodeVoting protects against vote manipulations at the voter's PC under the following assumptions: i) the CodeCard is generated in a secure and controlled environment by the VoterCard (the voter is the only person there), ii) the voter keeps her CodeCard secret, and iii) the correspondence between the candidate and its code (letter) cannot be changed. The last assumption can be achieved by publicly exposing the ballot or by any other technique that prevents a ballot change, such as using an image hard to forge/modify as a ballot.

Under these assumptions changing a vote to a predetermined candidate is virtually impossible because the corresponding vote code is not known by the attacker, and the probability of guessing the correct vote code, with 5 alphanumeric symbols, is 1 in 36^5, i.e. less than 1 in 60 million.

A random candidate change attack has almost the same probability of success as the previous attack. If we have n candidates running for the election the probability of guessing a random valid vote code is $n - 1$ in 36^5. However, to prevent an easy denial of service attack the CodeCard should not automatically block when the voter inserts invalid vote codes. Therefore, this limitation allows an attacker to perform a brute force attack to get a random valid vote code. To minimize such an attack simple measures can be used, such as delaying the vote code verification function and/or increasing the length of the vote codes. For instance, with a delay of three seconds the probability of a successful one hour brute force attack is just 1 in 50000, for a vote code with 5 alphanumeric symbols and $n = 11$. Moreover, adding to the three seconds delay an increase of the vote code length to 6 symbols results in a reduction of the attack's success probability to less than 1 in 1 million.

The other possible attack is to fool the voter into believing that she cast a vote, while in reality no vote was cast. To prevent such attack we propose the use of another code to confirm vote delivery. The VoterCard only releases this code after getting a confirmation that the vote was successfully delivered, e.g. could be a message signed by the election server. Therefore, if we use a confirmed vote delivery code with the same length of vote codes, this attack has the same probability of success than the attack of changing the vote to a predetermined candidate. The receipt received by the VoterCard can be stored inside of it to provide a poof of vote delivery, therefore allowing the voter to protest if her vote is not considered for the final tally.

If the voting system does not allow a voter to cast several votes there is also a possible attack to the voters' trust on the voting system. The attack goes as follows: the attacker lets the voter successfully cast a vote and then change the valid confirmed vote delivery code to a fake one. The voter would think that something wrong had happened, however there was nothing wrong. Then the voter would try to vote again and the voting system replies that the voter had already voted and, of course, the voter will protest. If the voting system allows the voter to cast several votes this attack would not be a problem.

Another valuable aspect to evaluate is the implication of CodeVoting in the vote buying/coercion problems. CodeVoting allows the voter to produce a receipt of the vote by giving away the CodeCard to an attacker prior to the election day. On election day, the attacker can demand the voter to vote using a computer controlled by the attacker, e.g. by using a web site controlled by the attacker or a special program develop by the attacker. In this case the attacker will have a vote receipt that proves the voter's vote, therefore enabling vote buying and coercion. We think the best way to prevent such attack is by allowing the voter to vote several times, i.e. update her vote. We believe that the possibility of a vote update in a machine not controlled by the attacker would discourage vote buying and coercion attacks.

7 CodeVoting Issues and Future Work

In this section we will present some issues we have identified with CodeVoting that we will address in future work, namely the capacity of dealing with a large candidate list and the problems raise by the reuse of the CodeCard.

Large candidate lists are a big issue when considering the real application of CodeVoting. As proposed, the CodeCard must have an entry for each possible candidate. If we consider elections with a large number of candidates, lets say above thirty, the size of the CodeCard starts to become to large and usability problems may arise.

Another issue is the CodeCard reuse. If a voter uses the same CodeCard in more than one election it is possible for an attacker to replace the voter's vote by a random one. To be able to perform this attack an attacker must collect the codes used on the previous elections. Additionally, the attacker must also be in control of the PCs used by the voter to vote in each election. If the voter uses

different PCs to vote it will be much harder to perform the attack. We can also make this attack harder by doing all the voting protocol inside the VoterCard, without leaking the voter's identification to the voter's PC. In this case the attacker only has the unlocking PIN to identify the voter.

Of course, the voter can always protect herself against the CodeCard reuse attack by getting a new CodeCard for her VoterCard between elections. However an issue still remains: simultaneous elections. The simultaneous elections issue is a particular case of the CodeCard reuse issue, in which the voter cannot go (or is not convenient to go) to the local authorities and get a new CodeCard. One solution that may work in some particular cases is to use sequential candidate labeling throughout all the simultaneous elections, e.g. instead of using candidate A and B for election 1 and 2, use candidate A and B for election 1 and candidate C and D for election 2. This simple candidate labeling solves the problem of simultaneous elections but can be lead to the large candidate list issue.

8 Conclusions

Automatic vote manipulation at client side is one of the biggest dangers that prevent the widespread of Internet voting. We present Code Voting a solution to prevent automatic vote manipulation at client side of a voting application that, at the same time, allows the use of cryptographic voting protocols that protect voters' anonymity and election's integrity at server side.

CodeVoting prevent the manipulation of the voter's vote at three levels: i) prevents the change of the vote to a predetermined candidate, ii) prevents the change of the vote to a random candidate, and iii) prevents vote dropping by a malicious voter's PC.

Besides providing protection against automatic vote manipulation, Code Voting can also provide additional protection to the vote and voter. Namely, the use of CodeVoting in conjunction with a cryptographic voting protocol that runs completely inside the VoterCard may prevent the vote client application, running on the voter's PC, from having access to the clear vote or a message that could directly identify the voter during the voting procedure, therefore providing additional protection to the voter's privacy at the client side of the voting system.

The use of CodeVoting also have implications concerning the vote buying/coercion problem, namely because the CodeCard can be used to get a receipt of the voter's vote. Nevertheless, these implications can be minimized if the voting system allows the voter to update her vote.

Another issues identified in CodeVoting that will deserve our attention in future work are the usability issues concerning elections with a large candidate list, and the issues related with the reuse of the CodeCard, specially in the case of simultaneous elections.

Acknowledgments

We would like to thank the anonymous reviewers and the conference participants for their helpful comments.

References

1. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: CCS 2004: Proceedings of the 11th ACM conference on Computer and communications security, NewYork, USA, pp. 132–145 (2004)
2. California Internet Voting Task Force: A report on the feasibility of Internet voting (January 2000), http://www.ss.ca.gov/executive/ivote
3. Chaum, David: SureVote. September 2007.// International patent WO 01/55940 A1 (02 August 2001), http://www.surevote.com/home.html
4. Clarkson, M., Myers, A.: Coercion-Resistant Remote Voting Using Decryption Mixes. In: Workshop on Frontiers in Electronic Elections, Milan, Italy (September 2005)
5. Estonian Internet Voting System (July 2007), http://www.vvk.ee
6. Hirt, M., Sako, K.: Efficient Receipt-Free Voting Based on Homomorphic Encryption. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 539–556. Springer, Heidelberg (2000)
7. Internet Policy Institute: Report of the National Workshop on Internet Voting: Issues and Research Agenda (March 2001), http://www.diggov.org/archive/library/dgo2000/dir/PDF/vote.pdf
8. Jefferson, D., Rubin, A., Simons, B., Wagner, D.: A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE) (January 2004), http://www.servesecurityreport.org/paper.pdf
9. Juels, A., Catalano, D., Jakobsson, M.: Coercion-Resistant Electronic Elections. In: Workshop on Privacy in the Electronic Society, Alexandria, Virginia, pp. 61–70 (November 2005)
10. Kutylowski, M., Zagórski, F.: Coercion-Free Internet Voting with Receipts. In: Workshop on e-Voting and e-Govrnment in the UK. Edinburgh (February 2006)
11. Lee, B., Kim, K.: Receipt-Free Electronic Voting Scheme with a Tamper-Resistant Randomizer. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 389–406. Springer, Heidelberg (2003)
12. Okamoto, T.: Receipt-Free Electronic Voting Schemes for Large Scale Elections. In: Security Protocols Workshop, Paris, France, pp. 25–35 (April 1997)
13. Rubin, A.: Security Considerations for Remote Electronic Voting Over the Internet. Communications of the ACM 45(12) (2002)
14. Sadeghi, A., Selhorst, M., Stüble, C., Wachsmann, C., Winandy, M.: TCG Inside? - A Note on TPM Specification Compliance. In: STC 2006: Proceedings of the 1st ACM Workshop on Scalable Trusted Computing, Virginia, USA (November 2006)
15. Sako, K., Kilian, J.: Receipt-Free Mix-Type Voting Scheme A Practical Solution to the Implementation of a Voting Booth. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 393–403. Springer, Heidelberg (1995)
16. UK's National Technical Authority for Information Assurance: e-Voting Security Study (July 2002), http://www.ictparliament.org/CDTunisi/ict_compendium/paesi/uk/uk54.pdf
17. Volkamer, M., Alkassar, A., Sadeghi, A., Schulz, S.: Enabling the Application of the Open Systems like PCs for Online Voting. In: FEE 2006: Proceedings of the Frontiers in Electronic Elections Workshop, Germany (September 2006)
18. Zúquete, A., Costa, C., Romao, M.: An Intrusion-tolerant e-Voting Client System. In: WRAITS 2007: 1st Workshop on Recent Advances on Intrusion-Tolerant Systems, Lisbon, Portugal (March 2007)

Author Index