Berner Fachhochschule
Technik und Informatik

Research on E-Voting Technologies

Dr. Rolf Haenni, Dr. Eric Dubuis, Dr. Ulrich Ultes-Nitsche

Research report BFH-TI Nr. 5 of 24th October 2008

The series „Research reports BFH-TI“ of the Berne University of Applied Sciences, Engineering and Information Technology gives an insight perspective of research and development at the BFH-TI.

Research reports published previously

Nr. 4 Authentication and Transaction Security in E-business. Dr. Lorenz Müller. January 2008.

Nr. 3 BFH-TI Forschungsauftrag 2006/2007 – Berichte der Forschenden. Dozierende der Berner FH Technik und Informatik. Januar 2008.

Nr. 2 BFH-TI Forschungsauftrag 2005/2006 – Berichte der Forschenden. Dozierende der Berner FH Technik und Informatik. November 2007.

Nr. 1 Action Cyphers – A new core component for E/D similar block ciphers. Dr. David-Olivier Jaquet-Chiffelle. Mars 2006.

Publisher
Berner Fachhochschule, Technik und Informatik, Postfach, CH-2501 Biel/Bienne
Forschung und Entwicklung, Seevorstadt 103b
T +41 (0)32 321 64 64, F +41 (0)32 321 65 65
technologietransfer@bfh.ch
ti.bfh.ch/forschung

Concept and execution
Jens Geisel

Editorial office
Author(s) and Jens Geisel

Berner Fachhochschule
Technik und Informatik

Research on E-Voting Technologies

Dr. Rolf Haenni, Dr. Eric Dubuis, Dr. Ulrich Ultes-Nitsche

Research report BFH-TI Nr. 5 of 24th October 2008

Abstract

There is an emerging demand on using the Internet for performing elections, votes, or polls. This paper provides an overview on recent e-voting technologies needed to carry out remote voting via the Internet. The survey starts with listing the most stringent security requirements for such systems. To satisfy these requirements, e-voting protocols involve several strong cryptographic primitives. This paper gives an overview of approaches based on blind signatures, anonymous channels, and homomorphic encryption. It also gives some references to formal verification techniques, which can be used to prove the correctness of a given e-voting protocol, and to risk analysis techniques, which can be applied to a given e-voting system to evaluate the risk of a successful attack. The survey concludes by mentioning the Swiss perspective on e-voting.

Key words

e-voting; internet-voting; requirements on e-voting systems; e-voting protocols; Swiss perspective

Biel/Bienne, October 2008

Authors

Prof. Dr. Rolf Haenni, Professor Fachbereich Informatik, Berner Fachhochschule
Prof. Dr. Eric Dubuis, Professor Fachbereich Informatik, Berner Fachhochschule
Prof. Dr. Ulrich Ultes-Nitsche, Departement Informatik, Universität Freiburg i. Ü.

Bern University of Applied Sciences

Research on E-Voting Technologies

A Survey

Rolf Haenni & Eric Dubuis
Bern University of Applied Sciences
CH-2501 Biel, Switzerland
{rolf.haenni,eric.dubuis}@bfh.ch

Ulrich Ultes-Nitsche
University of Fribourg
1700 Fribourg, Switzerland
uun@unifr.ch

October 24th, 2008

Contents

1 Introduction

Governments around the world are increasingly considering the replacement of traditional paper-based voting schemes with electronic voting systems. A particular form of such *e-voting* systems are those which allow voters to cast their ballots over the internet, so-called *remote e-voting* (or *i-voting*) systems.[1] In the last few years, several legally binding remote e-voting pilots have been conducted in various countries [10; 35], but most of them were restricted to communal or regional elections. The Swiss pilots in the cantons of Geneva, Neuchâtel, and Zürich are considered to be among the most advanced projects worldwide [9]. Despite the pioneering role of the Swiss e-voting projects, the first nationwide parliamentary elections took place in Estonia in February 2007 [38].

The idea of introducing electronic means into the electoral process has generated a lively debate, in which e-voting is viewed both a chance and a danger for democracy. The hope of e-voting enthusiasts includes the possibility of positive effects such as higher voter participation, improved pre-electoral opinion formation, or increased cost-effectiveness, whereas the fears of sceptics are mostly tied to security concerns and the resulting possibility of large-scaled frauds. The legitimacy of such security concerns has been demonstrated by the negative e-voting experience in the Netherlands, where all nationwide e-voting activities have been stopped in 2007 after the vulnerability of the deployed system had been exposed in public [37]. More recently, a group of some of the world's leading IT security experts has issued a statement warning that e-voting can not be verifiably accurate until "*serious, potentially insurmountable technical challenges*" are overcome.[2] The statement includes a list of technical challenges and the following general recommendation:

> [...] "*pilot studies of internet voting in government elections should be avoided, because the apparent success of such a study absolutely can not show the absence of problems that, by their nature, may go undetected.*"

From the perspective of this statements, and as the negative experience in the Netherlands has proved, it is crucial for an e-voting system to meet the highest security criteria before being introduced in practice.

2 Security Requirements

For an e-voting system to be secure, it has to function without vulnerabilities in potentially insecure environments such as the internet. For this, it has to be implemented according to a secure design. Despite the complexity of designing and implementing such a system, some criteria seem to be unanimously accepted as the core security requirements for e-voting [18; 43]:

Accuracy: A systems is accurate if casted votes can not be altered, validated votes can not be eliminated from the final tally, and invalid votes are not counted in the final tally.

[1] In this paper, we use the general term *e-voting* in a very restricted sense for remote e-voting over the internet.

[2] See http://verifiedvoting.org/downloads/InternetVotingStatement.pdf

Democracy: A system is democratic if only authorized voters can vote and eligible voters can only vote once.

Privacy: A system is private if no casted ballot can be linked to its voter (*anonymity*), neither by election authorities nor anyone else, and no voter can prove that he or she voted in a particular way (*receipt-freeness*).

Verifiability: A system is *individually* verifiable if voters can independently verify that their own votes have been counted correctly in the final tally. A system is *universally verifiable*, if voters can independently verify that all validated votes have been counted correctly in the final tally.

Fairness: A system is fair if no early results can be obtained before the voting period ends.

The literature on e-voting technologies offers various *protocols* to establish these core requirements (see Subsection 3.1). In the following, we will thus refer to them as *protocol requirements*. Note that some protocol requirements seem to be inherently contradictory, e.g. individual verifiability seems to be incompatible with receipt-freeness.

Further requirements, which address general security properties of an implemented system, are less specific to e-voting but still crucial for introducing remote e-voting in practice. Examples of such general *system requirements* are *availability*, *reliability*, *accountability*, *auditability*, *disclosability*, or *transparency* [42]. Note again that some system requirements seem to be in contradiction with some protocol requirements.

Apart from the above security criteria, there are some desirable properties such as *convenience*, *flexibility*, and *mobility*, which are influencing the efficiency and usability of an e-voting system, and are thus indirectly affecting the security of the election [18]. However, this paper will focus on the aforementioned protocol and system requirements. Further requirements, which address political, administrative, or juridical questions (for a detailed list of legal and operational standards as defined by the EU, see [16] or [59]), are also very important for introducing e-voting in practice, but the content of this paper is restricted to technical aspects of the problem.

3 Research on E-Voting Technologies

Research on remote e-voting has many forms and different facets. Roughly speaking, it is based on two pillars, a technical and a methodological one, and involves two layers, a theoretical and a practical one. The technical pillar includes basic techniques from areas such as cryptography, protocol design, IT and network security, or web technology, whereas the methodological pillar consists of formal verification methods, security analysis schemes, or attack models. The theoretical layer addresses questions related to the design and the verification of voting protocols (to achieve the above-mentioned protocol requirements), whereas the practical layer is devoted towards the implementation and analysis of concrete systems (to achieve the above-mentioned system requirements). Note that the two layers are intrinsically interwoven.

According to this general classification and as illustrated in Fig. 1, research on e-voting technologies can be divided into the following four areas:

1. Design of e-voting protocols;
2. Verification of e-voting protocols;
3. Implementation of e-voting systems;
4. Analysis of e-voting systems.

In the remaining of this section, a short research survey is given for each these areas.

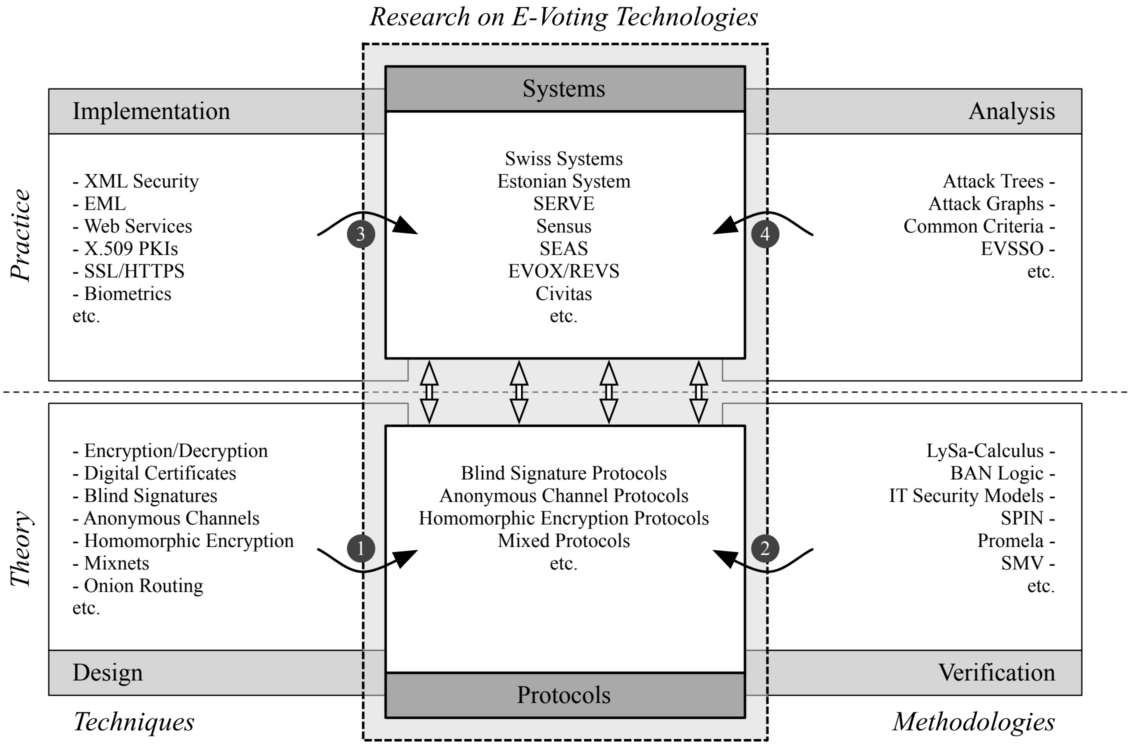

Figure 1: Overview and classification of research in e-voting technologies.

3.1 Designing E-Voting Protocols

The design of a secure e-voting protocol usually involves several strong cryptographic primitives. Besides the usual application of symmetric encryption to establish confidential channels, asymmetric encryption to exchange session keys, and digital signatures and certificates to ensure the integrity and authenticity of the transmitted messages, there are at least three specific design approaches for building e-voting protocols based on strong cryptography: protocols based on *blind signatures* [22], *anonymous channels* [12], and *homomorphic encryption*[7], respectively. The key ideas of those techniques are the following:

Blind Signatures: The largest family of voting-oriented protocols is based on the concept of a blind (digital) signature [13]. The idea is to apply a blinding function to a message before sending it to a signing party. Based on a simple RSA-like scheme, this can be done such that the blinding function can be inverted on the blinded signature to finally obtain a regular RSA-based digital signature. At the end of this process, in which the content of the message has been entirely disguised from the signer, the signature is ordinarily verifiable using the signer's public key.

Applying blind signatures to e-voting has first been proposed in [22]. In the suggested protocol, the blind signature is used to detach a validated ballot from the voter's identity without requiring anonymous channels. Together with some other cryptographic primitives, e.g. encryption or bit commitment, all protocol requirements except receipt-freeness (privacy) are guaranteed. A major drawback of the protocol is that voters needs to be active in at least two phases to ensure verifiability and fairness, which restricts its usability intrinsically. To overcome these drawbacks, many variations of this protocol have been suggested in the literature [5; 44; 45; 49], and many prototype implementations have been realized (see Subsection 3.3). With slight modifications, it is still generally regarded as one of the best voting protocols. It is simple, flexible, and efficient.

Anonymous Channels: Another large family of e-voting protocols are based on anonymous channels [12; 20]. They allow messages to be sent anonymously, i.e. such that a recipient can not trace the received messages back to the senders. The most prominent technique for building anonymous channels are *mixnets* [14] and *onions* [25]. They are comprised of a collection of *anonymization servers* whose task is to shuffle a given input sequence of encrypted messages. To ensure that mix-servers or onion routers do not drop or substitute messages, it is necessary that the servers provide proofs of correct operation. The resulting anonymous channel is then called *verifiable*. Although most existing verifiable anonymous channels are relatively inefficient, progress has been reported recently [23; 41].

Similarly to blind signatures, anonymous channels are often used as cryptographic primitives to provide anonymity in e-voting protocols [41; 48] and implementations thereof (see Subsection 3.3). Some of them satisfy almost all protocol requirements, but guaranteeing receipt-freeness together with verifiability is again very challenging [36].

Homomorphic Encryption: A third category of e-voting protocols is based on what is known as *homomorphic encryption* [7]. The idea here is to apply an arithmetic operation, let's say addition, to encrypted numbers without previously decrypting them.[3] The resulting encrypted sum can then be revealed with a private decryption key. Homomorphic encryption schemes ensure that individual numbers can not be decrypted with the private decryption key only.

In a voting schemes based on homomorphic encryption, voters may openly authenticate themselves to the voting servers. As no individual vote ever needs to be revealed, there is no need for blind signatures or anonymous channels to ensure voter privacy.

[3] In abstract algebra, a function with such a structure-preserving property is called a *homomorphism*.

To prove the validity of each ballot in the voting stage, homomorphic schemes require computationally intensive zero-knowledge proofs. Another drawback is that homomorphic approaches do not allow complex multi-candidate elections. Most of them are restricted to simple yes/no votes which are easy to count. In the literature, various protcols are based on homomorphic encryption [17; 53], but only few concrete implementations exist. Note that quite some research on receipt-freeness has been conducted in this approach [29].

With respect to the suggested catalog of protocol (and system) requirements, each of the above approaches has its own advantages and disadvantages. To overcome the disadvantages of a given approach, some protocols try to mix elements of different approaches. Note that the protocols of some concrete implementations do not fall into these categories as they do not employ any of the voting-oriented strong cryptographic primitives (see Subsection 3.3).

3.2 Verifying e-Voting Protocols

Verification of e-voting protocols is at a first glance not very different from protocol verification in general. And there exist well-established protocol-verification techniques. Most of these techniques and their implementations in verification tools are aiming to be fully automatic and therefore use finite-state verification techniques. These verification techniques and tools include (just to name a few of the most important ones):

- SPIN and Promela,[4]
- SMV,[5]
- FDR.[6]

The above mentioned verification tools are more or less general purpose verifiers. As e-voting poses specific security challenges, some people tried to consider specific security-focussed verification systems. The following approaches are worth mentioning:

- LySa-Calculus [43];
- IT-security models [26; 40], e.g. the *Integrity Model* [15] or *Confidentiality Model* [6];
- BAN-Logic [11] (for authentication).

Having discussed automated verification approaches up to now, one should keep in mind that the discussed techniques concentrate on the *protocol* aspects of verification; the proper functioning of the involved cryptographic functions is simply assumed. This may not be sufficient when dealing with e-voting protocols. Therefore it is worth mentioning here that there exist semi-automatic approaches based on proof systems. Examples of such interactive approaches are:

[4] See project web site at http://spinroot.com.

[5] See project web site at http://www.cs.cmu.edu/ modelcheck/smv.html.

[6] See project web site at http://www.fsel.com/software.html.

- HOL/Isabelle,[7]
- PVS.[8]

Considering proof systems for the verification of e-voting protocols allows to verify the cryptographic functions involved in the protocols in the context they are applied in the protocol. This is important since the usage of secure cryptographic functions in the context of particular protocols can render the cryptographic functions insecure (an example is the security protocol WEP used in IEEE 802.11 WLANs, which uses the perfectly secure RC4 symmetric-key cipher in such a way that symmetries in the usage of RC4 allow to calculate the symmetric key [57]).

3.3 Implementing E-Voting Systems

Various e-voting systems are in operation today in many parts of the world. Some of them were developed by specialized companies and are available as commercial products, while others are customized solutions for individual operators. In Switzerland, for example, the system deployed in Neuchâtel is a commercial product from a vendor in Spain, while the existing systems in Zürich and Geneva are individual developments. A comprehensive, world-spanning overview of the pilots and test runs is available through the *E-Voting Database*,[9] which is maintained by the *Competence Center for Electronic Voting and Participation* in Austria. Note that there is also a huge private market for e-voting systems beyond the public sector of political elections.

From an academic point of view, e-voting systems should always be built on top of one of the most sophisticated protocols from the e-voting literature (see Subsection 3.1). The idea is to achieve the core security requirements intrinsically as specific properties of the protocol, i.e., without relying on too many defeasible assumptions. Another general requirement, which is often mentioned in academic discussions on e-voting, is to provide a maximum level of transparency and openness. There are even some private movements which promote publicly owned and administered voting systems based on open-source software.[10] They argue that transparency and openness are the key principles to guarantee citizen's confidence in the electoral process [3; 34]. Note that most of the operational systems deployed for legally binding elections are either not very transparent or are not based on any of the latest protocols from the e-voting literature.

Along with the development of commercial and individual e-voting solutions, there is a number a prototype implementations of some of the most well-known protocols. Most notably, there are several implementations of blind signature schemes based on the original protocol from [22] (see Subsection 3.1). One of the first and most-cited prototype systems is Sensus, which has been implemented and tested at the Washington University [19]. Similar implementations are Evox, which has been used for campus-wide elections at the MIT [21; 28], and Votopia, which has been built for the FIFA WorldCup 2002 in Korea/Japan to select the top 10 most valuable players and the best goal keepers [33].

[7] See project web site at http://www.cl.cam.ac.uk/research/hvg/Isabelle.

[8] See project web site at http://pvs.csl.sri.com.

[9] To access the database, go to http://db.e-voting.cc.

[10] For example, see http://www.openvotingconsortium.org.

More recent implementations of the same type of blind-signature based protocols are Revs [31], Seas [4], and one without a particular name [2]. Recently, the protocol proposed in [49] has been implemented as a Bachelor thesis at the Bern University of Applied Sciences under the supervision of one of the applicants [1]. There are also various prototype and commercial implementations of schemes based on anonymous channels (e.g. VoteHere VHTi,[11] Scytl Pnyx [50], or SureVote[12]) or homomorphic encryption (e.g. Adder [32] or CyberVote[13]).

Beside the efforts of building prototype systems as part of academic research projects, there are also some movements towards the standardisation of general technical means for e-voting processes. The most prominent advances of that kind is the specification of the *Election Markup Language* (EML). This is a XML-based OASIS standard for exchanging various types of election-related messages during nomination, voter registration, voting, and counting [8]. EML is based on a high-level definition of the entire electoral process, and is thus not tied to a specific protocol or a particular type of underlying network. In addition to such standardisation attempts, there is also a number of initiatives which try to provide reference implementations of e-voting systems as open-source software. One of them is the Emv2003 project of the *Open Voting Consortium*.[14] Note that EMV2003 is not based on any of the voting-oriented cryptographic primitives mentioned in Subsection 3.1. Other systems that fall into this categorie of free non-academic software are CIVS[15] and GNU.FREE.[16]

In addition to the above attempts of implementing secure and robust e-voting systems, there is also a number of very general security issues, which become particularly important in e-voting applications. One of them is the so-called *secure platform problem*, which refers to the problem of protecting an inherently insecure client-side platform against malicious software and corresponding attacks [24]. In remote e-voting over the internet, this type of vulnerability is always a major concern. To overcome this problem, specific techniques such as *code voting* have been suggested [27; 47]. A more general solution would be to employ a trusted computing environment, as suggested in [51].

Other general security issues, e.g. voter authentication or channel confidentiality, are typically solved using common cryptographic techniques (encryption, certificates, PKI, hash functions, etc.) and corresponding technologies (AES, SSL, X.509, etc.), or with common practices (PIN/TAN, CAPTCHA, biometrics, etc.).

3.4 Analysing E-Voting Systems

For a given implementation of an e-voting system, the immediate question is whether all its components are such that the overall security of the system is guaranteed. Note that analyzing an implemented system is quite different from verifying its underlying protocol (see Subsection 3.2). In computer security, a typical approach for such an analysis is to define a *threat model*, which defines a set of possible attacks to consider. A detailed threat

[11] See project web site at http://www.votehere.net/old/vhti.php.
[12] See project web site at http://www.surevote.com.
[13] See project web site at http://www.eucybervote.org
[14] See project web site at http://evm2003.sourceforge.net.
[15] See project web site at http://www.cs.cornell.edu/andru/civs.html.
[16] See project web site at http://www.j-dom.org/users/re.html.

model specifies for each possible attack the probability of the attack to happen and the potential damage. The model can then be used to evaluate to overall risk of using the system or to locate its weaknesses. Two of the most prominent formal threat modeling techniques are the *attack trees* and *attack graphs*, but there are also less formal approaches such as the *Common Criteria* standard.

Attack Trees: Attack trees are a graphical means for the investigation of the security of a computer system [52; 54]. An attack tree consists of one root, leaves, and children. From the bottom up, child nodes are conditions which must be satisfied to make the direct parent node true; when the root is satisfied, the attack is complete. Each node may be satisfied only by its direct child nodes. Note that attack trees can become largely complex, especially when dealing with specific attacks. A full attack tree may contain hundreds or thousands of different paths all completing the attack. Even so, these trees are very useful for determining which threats exist and how to deal with them.

Attack Graphs: An attack graph is a set of actions that increase an adversary's capabilities [56; 60]. The graph can focus on whether a certain set of initial capabilities can eventually lead to some critical capability, or it can focus on the extent to which an adversary can penetrate a network given an initial set of capabilities. Traditional graph-based analyses can be applied to identify optimal changes to the network.

Common Criteria: The Common Criteria (CC) is an international standard for information technology security evaluation.[17] The CC provides a framework for users to specify their security requirements in a so called *protection profile*, for manufacturers to make claims about the security properties of their products, and for testing laboratories to evaluate the products. Based on the evaluation report the certification authority can decide whether the company gets a certificate for its product.

In the e-voting literature, papers on security analysis based on threat models are still very rare. One of the few general security optimization methods is called *EVSSO* (E-Voting System Sycurity Optimization) [46]. It was developed to evaluate and to measure the security of e-voting systems. This method points out security flaws of an examined system, shows its security optimization potential, and can be used to compare different electronic voting systems. The methodology differs from other approaches insofar as it is a holistic approach and takes the interdependencies of different aspects of the voting system into account. It visualizes the security situation of an e-voting system in a clear way and shows its potential for improvement.

Another very general approach is the application of the Common Criteria standard to e-voting systems. The proposed method in [59] is based on a very detailed catalog of e-voting-specific requirements, from which a corresponding CC protection profile has been developed [26]. In [58], approaches of that kind are called *taxonomy check-lists*.

In a more specific study [39], multi-parameter attack trees have been used to compare the security of two concrete e-voting systems: the Estonian e-voting system and the system

[17] See project web site at http://www.commoncriteriaportal.org.

SERVE (Secure Electronic Registration and Voting Experiment) developed in the USA.[18] For proving the practical security of the systems, the approach is based on a very detailed *environment model*, which includes society characteristics, security assumptions, and properties of possible adversaries. Based on the underlying attack tree, it has been shown the Estonian system is much more resistant against large-scale attacks than SERVE.

4 The Swiss Perspective

The Swiss government has repeatedly declared its strong commitment to introducing e-voting technologies in Switzerland, and has assigned the responsibility of supervising the cantons in their efforts of conducting pilot projects to the Federal Chancellery. As the negative experience in the Netherlands has shown, it is a matter of utmost importance for the future democratic processes in our country that these developments are conducted into the right direction. This remark applies to the technological as well as the political-juridical side of the issue. With respect to technological questions, it is important that developers and operators of e-voting systems keep track of the latest findings in academic research, and that these findings find their way into the deployed systems. For this to happen, it is important to establish strong links between the persons and institutions in charge of introducing e-voting and researchers with the competence of judging the relevance of those findings.

Of particular significance for future e-voting research in Switzerland is to give some special attention to the particular constitutional, political, and legal situation. While the general security requirements of an e-voting system are totally independent of the situation of a particular country (see Section 2), it is possible that one or the other requirement is less (or more) difficult to achieve within the given constraints in a particular country. The following list gives some examples of such Swiss particularities:

- Postal voting is established and accepted (unlike e.g. France);
- Citizen do not receive digital IDs (unlike e.g. Estonia);
- Multiple significant parties are involved (unlike e.g. USA);
- Frequent initiatives and referenda require simple yes/no ballots (unlike most other countries);
- Decentralized voter registers are administered by local municipalities (unlike many other countries).

Some of these particularities may favor certain approaches or techniques. For eaxmple, a multi-party system may help to remove the residual risk in a mixnet approach with multiple trusted parties, whereas simple yes/no ballots in initiatives and referenda may simplify the secure platform problem and are predestined for homomorphic voting schemes.

[18] SERVE was planned for deployment in the 2004 primary and general elections, but following the recommendation of a group of security experts, all the project activities were stopped beforehand [30].

5 Conclusion

Switzerland is one of the most ambitious and most advanced countries in introducing remote e-voting. For the long-term success of e-voting in Switzerland and abroad, it is crucial to employ the highest possible security measures. The scenario of a successful large-scale attack against real elections is threatening, it would shatter people's confidence in our democratic system at its core. While considerable research progress has been achieved in designing and implementing secure e-voting systems, there is still no common agreement on how the ideal system should look like. This lack of consensus is a reason for many academics to be skeptical with respect to the security offered e.g. by commercial systems. As the negative e-voting experience in the Netherlands has demonstrated [37], such skepticism is often more than legitimate.

Many security-critical commercial applications are widely accepted among users today. But since there is a number of fundamental differences between e-commerce and e-voting [30; 55], it is obviously a mistake to assume that just because commercial transactions can be safely conducted over the internet, it is also possible to cast votes safely over the internet using the same mechanisms. As there are still many open security problems particularly related to remote e-voting, further theoretical and practical research is necessary before further introducing e-voting systems in practice.

References

[1] A. Aeby and M. Wiget. On-Line Meinungsumfragen. Diploma thesis, Bern University of Applied Sciences, Biel, Switzerland, 2007.

[2] R. Anane, R. Freeland, and G. Theodoropoulos. e-voting requirements and implementation. In *CEC'07, 9th IEEE Conference on E-Commerce Technology*, pages 382–392, Tokyo, Japan, 2007.

[3] D. Ataç, B. Kayapinar, and W. Riedel. e-Voting mit Open Source. Gutachten, Informatik und Gesellschaft, Technische Universität Berlin, 2004.

[4] F. Baiardi, A. Falleni, R. Granchi, F. Martinelli, M. Petrocchi, and A. Vaccarelli. SEAS, a secure e-voting protocol: Design and implementation. *Computers & Security*, 24(8):642–652, 2005.

[5] A. Baraani-Dastjerdi, J. Pieprzyk, and R. Safavi-Naini. A practical electronic voting protocol using threshold schemes. Technical report, University of Wollongong, Department of Computer Science, Wollongong, Australia, 1994.

[6] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations. Technical Report MTR-2547, The MITRE Corporation, Bedford, USA, 1973.

[7] J. D. C. Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, New Haven, USA, 1987.

[8] J. Borras. Election markup language (EML) v5.0: Process and data requirements. Committee Specification 01, OASIS Election and Voter Services TC, 2007.

[9] N. Braun and D. Brändli. Swiss e-voting pilot projects: Evaluation, situation analysis and how to proceed. In R. Krimmer, editor, *2nd International Workshop on Electronic Voting*, number P-86 in Lecture Notes in Informatics, pages 27–36, Bregenz, Austria, 2006. Gesellschaft für Informatik E.V.

[10] T. M. Buchsbaum. E-voting: International developments and lessons learnt. In R. Krimmer, editor, *1nd International Workshop on Electronic Voting*, number P-47 in Lecture Notes in Informatics, pages 31–42, Bregenz, Austria, 2004. Gesellschaft für Informatik E.V.

[11] M. Burows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[12] D. Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. *Communications of the ACM*, 24(2):84—88, 1981.

[13] D. Chaum. Blind signatures for untraceable payments. In *CRYPTO'82, 2nd International Cryptology Conference*, pages 199–203, Santa Barbara, USA, 1982.

[14] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[15] D. C. Clark and D. R. Wilson. A comparison of commercial and military security policies. In *IEEE Security and Privacy Symposium*, pages 184–194, Oakland, USA, 1987.

[16] Council of Europe. *Legal, Operational and Technical Standards for e-Voting.* Rec(2004)11. Council of Europe Publishing, 2004.

[17] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. *European Transactions on Telecommunications*, 8(5):481–490, 1997.

[18] L. F. Cranor and R. K. Cytron. Design and implementation of a practical security-conscious electronic polling system. Technical Report WUCS-96-02, Washington University, 1996.

[19] L. F. Cranor and R. K. Cytron. Sensus: A security-conscious electronic polling system for the internet. In *HICSS-30, 30th Hawaii International Conference on System Sciences*, volume 03, pages 561–570, Maui, USA, 1997.

[20] G. Danezis and C. Diaz. A survey of anonymous communication channels. *Journal of Privacy Technology*, 2008 (submitted).

[21] B. W. DuRette. Multiple administrators for electronic voting. Bachelor thesis, Massachusetts Institute of Technology, Boston, USA, 1999.

[22] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In J. Seberry and Y. Zheng, editors, *ASIACRYPT'92, Workshop on the Theory and Application of Cryptographic Techniques*, LNCS 718, pages 244–251, Gold Coast, Australia, 1992.

[23] J. Furukawa and K. Sako. An efficient scheme for proving a shuffle. In J. Kilian, editor, *CRYPTO'01, 21st Annual International Cryptology Conference on Advances in Cryptology*, LNCS 2139, pages 368–387, Santa Barbara, USA, 2001.

[24] E. Gerck, C. A. Neff, R. L. Rivest, A. D. Rubin, and M. Yung. The business of electronic voting. In P. F. Syverson, editor, *FC'01, 5th International Conference on Financial Cryptography*, LNCS 2339, pages 243—268, Grand Cayman, British West Indies, 2001.

[25] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private Internet connections. *Communications of the ACM*, 42(2):39–41, 1999.

[26] R. Grimm and M. Volkamer. Development of a formal IT-security model for remote electronic voting systems. In R. Krimmer and R. Grimm, editors, *3nd International Workshop on Electronic Voting*, Lecture Notes in Informatics, pages 185–196, Bregenz, Austria, 2008. Gesellschaft für Informatik E.V.

[27] J. Helbach and J. Schwenk. Secure internet voting with code sheets. In A. Alkassar and M. Volkamer, editors, *VOTE-ID'07, 1st International Conference on E-Voting and Identity*, LNCS 4896, pages 166–177, Bochum, Germany, 2007.

[28] M. A. Herschberg. Secure electronic voting using the world wide web. Master's thesis, Massachusetts Institute of Technology, Boston, USA, 1997.

[29] M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *EUROCRYPT'00, International Conference on the Theory and Applications of Cryptographic Techniques*, LNCS 1807, pages 539–556, Bruges, Belgium, 2000.

[30] D. Jefferson, A. D. Rubin, B. Simons, and D. Wagner. A security analysis of the secure electronic registration and voting experiment (SERVE). Technical report, 2004.

[31] R. Joaquim, A. Zuquete, and P. Ferreira. REVS – a robust electronic voting system. In *IADIS International Conference e-Society 2003*, pages 95–103, Lisbon, Portugal, 2003.

[32] A. Kiayias, M. Korman, and D. Walluck. An internet voting system supporting user privacy. In *ACSAC'06, 22nd Annual Computer Security Applications Conference*, pages 165–174, Miami Beach, USA, 2006.

[33] K. Kim. Killer application of PKI to internet voting. In *IWAP'02, 2nd International Workshop for Asia Public Key Infrastructures*, Taipei, Taiwan, 2002.

[34] J. Kitcat. Source availability and e-voting: an advocate recants. *Communications of the ACM*, 47(10):65–67, 2004.

[35] R. Krimmer, S. Triessnig, and M. Volkamer. The development of remote e-voting around the world: A review of roads and directions. In A. Alkassar and M. Volkamer, editors, *VOTE-ID'07, 1st International Conference on E-Voting and Identity*, LNCS 4896, pages 1–15, Bochum, Germany, 2007. Springer.

[36] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Providing receipt-freeness in mixnet-based voting protocols. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *ICISC'03, 6th International Conference on Information Security and Cryptology*, LNCS 2971, pages 245–258, Seoul, Korea, 2003.

[37] L. Loeber. E-voting in the Netherlands: from general acceptance to general doubt in two years. In R. Krimmer and R. Grimm, editors, *3nd International Workshop on Electronic Voting*, Lecture Notes in Informatics, pages 21–30, Bregenz, Austria, 2008. Gesellschaft für Informatik E.V.

[38] E. Maaten and T. Hall. Improving the transparency of remote e-voting: The estonian experience. In R. Krimmer and R. Grimm, editors, *3nd International Workshop on Electronic Voting*, Lecture Notes in Informatics, pages 31–44, Bregenz, Austria, 2008. Gesellschaft für Informatik E.V.

[39] T. Mägi. Practical security analysis of e-voting systems. Master's thesis, Tallinn University of Technology, Tallinn, Estonia, 2007.

[40] J. McLean. The specification and modeling of computer security. *IEEE Computer*, 23:9–16, 1990.

[41] C. A. Neff. A verifiable secret shuffle and its application to e-voting. In P. Samarati, editor, *CCS'01, 8th ACM Conference on Computer and Communications Security*, pages 116–125, Philadelphia, USA, 2001.

[42] P. G. Neumann. Security criteria for electronic voting. In *NCSC'93, 16th National Computer Security Conference*, pages 478–482, Baltimore, USA, 1993.

[43] C. R. Nielsen, E. H. Andersen, and H. R. Nielson. Static validation of a voting protocol. *Electronic Notes in Theoretical Computer Science*, 135(1):115–134, 2005.

[44] M. Ohkubo, F. Miura, M. Abe, A. Fujioka, and T. Okamoto. An improvement on a practical secret voting scheme. In M. Mambo and Y. Zheng, editors, *ISW'99, 2nd International Workshop on Information Security*, LNCS 1729, pages 225–234, Kuala Lumpur, Malaysia, 1999.

[45] T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In B. Christianson, B. Crispo, T. M. A. Lomas, and M. Roe, editors, *5th International Security Protocols Workshop*, LNCS 1361, pages 25–35, Paris, France, 1997.

[46] B. Ondrisek. *Sicherheit elektronischer Wahlen.* PhD thesis, Technische Universität Wien, Austria, 2008.

[47] R. Oppliger. How to address the secure platform problem for remote internet voting. In *SIS'02, 5th Conference on "Sicherheit in Informationssystemen"*, pages 153–173, Vienna, Austria, 2002.

[48] C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In T. Helleseth, editor, *EUROCRYPT'93, Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, LNCS 765, pages 248–259, Lofthus, Norway, 1993.

[49] I. Ray, I. Ray, and N. Narasimhamurthi. An anonymous electronic voting protocol for voting over the internet. In *WECWIS'01, 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pages 188–191, San Jose, USA, 2001.

[50] A. Riera and P. Brown. Bringing confidence to electronic voting. *Electronic Journal of e-Government*, 2(1), 2004.

[51] A. D. Rubin. Security considerations for remote electronic voting. *Communications of the ACM*, 45(12):39–44, 2002.

[52] V. Saini, Q. Duan, and V. Paruchuri. Threat modeling using attack trees. *Journal of Computing Sciences in Colleges*, 23(4):124–131, 2008.

[53] K. Sako and J. Kilian. Secure voting using partially compatible homomorphisms. In Y. Desmedt, editor, *CRYPTO'94, 14th Annual International Cryptology Conference on Advances in Cryptology*, LNCS 839, pages 411–424, Santa Barbara, USA, 1994.

[54] B. Schneier. Attack trees: Modeling security threats. *Dr. Dobb's Journal*, 24(12):21–29, 1999.

[55] G. Schryen. How security problems can compromise remote internet voting systems. In A. Prosser and R. Krimmer, editors, *1nd International Workshop on Electronic Voting*, number P-47 in Lecture Notes in Informatics, pages 121–131, Bregenz, Austria, 2004. Gesellschaft für Informatik E.V.

[56] O. M. Sheyner. *Scenario Graphs and Attack Graphs.* PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 2004.

[57] E. Tews, R. P. Weinmann, and A. Pyshkin. Breaking 104 bit WEP in less than 60 seconds. Cryptology ePrint Archive, Report 2007/120, 2007.

[58] T. Tjøstheim, T. Peacock, and P. Y. A. Ryan. A model for system-based analysis of voting systems. In *15th International Workshop on Security Protocols*, pages 77–95, Brno, Czech Republic, 2007.

[59] M. Volkamer and M. McGaley. Requirements and evaluation procedures for evoting. In *ARES'07, 2nd International Conference on Availability, Reliability and Security*, pages 895–902, Vienna, Austria, 2007.

[60] J. M. Wing. Attack graph generation and analysis. In *ASIACCS'06, ACM Symposium on Information, Computer and Communications Security*, pages 14–14, Taipei, Taiwan, 2006.