View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by KITopen

From Legal Principles to an Internet Voting System

Melanie Volkamer, Dieter Hutter

German Research Center for Artificial Intelligence GmbH
DFKI Saarbrücken
66123 Saarbrücken, GERMANY
volkamer@dfki.de, hutter@dfki.de

Abstract: Past research on Internet voting has been concentrated on two aspects. First, there are investigations to find the appropriate balance between anonymity and authentication. Second, the impact of the use of Internet voting to legislation has been studied. In this paper we analyze the impact of legislation to the design of a real Internet voting system. We discuss how legal aspects constitute security requirements on a technical level and refine the security requirements on the design level to corresponding security requirements of the resulting system.

1 Introduction

Reforms of the execution of democratic elections have taken place several times in the past. In the advent of e-democracy and e-government initiatives, the question arose whether and how citizens can be entitled to use the Internet in order to participate in elections. In the last years various voting systems, like for instance the i-vote system [FGr] in Germany, have been developed and tested in various countries. The popularity of Internet voting reached its peak in 2001. However, at the same time the difficulties in developing a legal voting system satisfying the required security properties have become obvious.

There are various proposed approaches for Internet voting (see [Sch96] for an introduction). We distinguish between Internet voting systems using polling stations and those allowing the voters to use their own personal equipment. With respect to the authentication to the system, a voter can legitimate herself either by presenting her PIN (or TAN) codes or by using an existing digital signature infrastructure. Systems also differ in the characteristics of the components an user has to trust in when using the system or they differ in the used cryptographic algorithm.

Since voting systems are complex distributed systems, it is rather difficult to understand up to what degree the system will guarantee the required security properties. Furthermore, up to now there are no standard criteria available, like for instance a Common Criteria Protection Profile [ISO00], to evaluate and certify Internet voting systems.

That is why developing an Internet voting system that is accepted by the voters and that also satisfies all requirements in a traceable way is still an unsolved task.

In this paper we use the basic methodology of the Common Criteria to develop technical requirements for a suitable voting system from the given legal preconditions that are formulated in electoral laws and constitutions. We start with the discussion of the legal principles in chapter 2 and develop a trust model based on these legal principles in chapter 3. Using this model we deduce compulsory requirements for the system design in chapter 4. In chapter 5, we present our Internet voting system *SecVote* and investigate in the next step the mechanisms to meet all requirements set up by the trust model. Finally, chapter 6 gives some details about the implementation of this system.

2 Legal Principles

The touchstone in developing an Internet voting system is represented by the necessity to meet the requirements of legal principles ([Wil02] for an introduction). In Germany, like in many other democracies, all elections have to satisfy basic voting principles which are formulated in constitutions and electoral laws. Elections have to be **universal, equal, free, secret** and **direct**.

The principle of **universal** elections guarantees equal suffrage for everybody which also means equal access to voting. For instance, it is not allowed to exclude any persons subgroups from an election. **Equal** elections guarantee that all ballots have the same influence on the result. Furthermore, voters are able to vote in the same formal way. The principle of **free** elections requires the facility for every voter to cast her ballot free of duress and without unlawful and undue influence. In particular this implies that a voting system must anticipate that a voter can be influenced by leaking intermediate results of an ongoing election. **Secrecy** of elections demands that only the voter is aware of her voting decision, which may never revealed to anybody else without her permission. To prevent disposal of votes the voter must not be able to prove anybody the result of her voting. The principle of **direct** elections prevents someone from voting on behalf of other eligible voters or the use of an electoral college.

3 Trust Model

In this chapter we derive the trust model from the legal principles presented above. We assume two groups of persons interacting with the voting system. First there are people who are interested in the correctness and security of the system: “honest” voters using the system and the organizers of the election maintaining the system. Second there is a malicious attacker who might be also camouflaged as a voter or an organizer.

We assume that this attacker is very powerful: He is able to read, save and delete all protocol messages - especially all transmitted ballots. The attacker can generate new messages or modify intercepted messages and send them to arbitrary system components. He is computationally restricted with respect to his computing resources during the election but we act on the assumption that an attacker might be able to overcome this restriction in the future. The attacker can also observe who actually is in the polling station at a given point of time. Equipped with these abilities, he tries to corrupt the secrecy of the votes of specific individuals, to manipulate the result of the election or simply to obstruct the election in general.

Honest groups act in compliance with the rules of the voting system and assist in detecting any kind of election fraud. These participants have two kinds of requirements: system requirements and those to the environment. So we developed an Internet voting system satisfying the legal principles if environmental requirements are guaranteed.

3.1 Requirements to the system

In the following we derive the **system requirements** of a voting system by analyzing the legal principles more closely:

The principle of **universal** election requires that the voting system is available for all voters independent of their personal holdings, can be used by all voters without requiring special knowledge, for instance in computer science, does not lose any data (e.g. during ballot transmission), and counts all ballots correctly.

Availability of the voting system implies that it must never enter an undefined state and that there is a trustworthy backup mechanism to recover the system in case of an emergency, e.g. a hardware failure.

The principle of **equal** election results in the need to prevent unauthorized access to the system. Voters have to authenticate themselves, each person can only vote at most once, and each ballot is counted exactly once within the result. As a consequence attackers must not be able to modify, copy or generate ballots without being detected by the organizers.

The principle of **free** voting means that attackers must not be able to influence a voter's decision which implies that it must be impossible to observe the voter in her decision. Also voters must not be able to prove their own decision to someone else because otherwise they might sell their votes. Until the election deadline is reached, the ballots must be transmitted and saved confidentially to prevent the calculation and publication of intermediate results.

The principle of **secret** election requires that any mapping of a voter to her ballot must be impossible during the election but also for the future. We have to take into account that both, the computational resources as well as the knowledge on cryptography will steadily increase in the future.

This requirement will essentially influence the design of ballot transmission and storage. The principle of secret election is an essential precondition for free voting.

There is no technical proviso for Internet voting with respect to the principle of **direct** elections.

Summing up, there are far more requirements arising from legal principles than ensuring secrecy and integrity of individual votes as it is often mentioned. Furthermore it is important to notice that the secrecy of election must be unconditionally ensured forever regardless of ongoing technological improvements.

3.2 Preconditions to the environment

Internet voting systems are technical systems which will only operate correctly if the environment is able to guarantee certain preconditions. For example, software systems requires dependable hardware which itself depends on a reliable power supply. Analogously, we have to assume certain preconditions on the environment in which the voting system will run to ensure the security of the overall system.

We assume that an attacker will only be able to manipulate a single component of the voting system. Our approach has to guarantee that the malicious corruption of a single component will be either detected during the election or else will not inflict the security of the system. The rationality behind this assumption is that the different components will be distributed on different locations and different persons will be in charge to maintain and supervise them. So we assume that organizational means will make sure that persons in different positions and locations will not collaborate in corrupting the system. Additionally we also suppose that people from different lobbies, who share a secret, do not work together to manipulate the election (principle of separation of functions and dual control). Furthermore, we assume that more than one voter casts her vote and not all votes are identical. Moreover we suppose that not all voters apart from one will conspire against the remaining voter to find out her decision.

Additional requirements are that the components are secure platforms (e.g. using a secure Linux version only equipped with the voting software) because otherwise we would have to trust in all other installed software and there might be a lot of possible attacks caused by Trojan horses. Such a program could cast the vote without voter's knowledge or it could even change the voter's decision before sending the ballot. Another possibility would be that the Trojan horse would send the voter decision directly to the attacker. Consequently the attacker reaches his goals independent from the system architecture and the used protocols.

Having these requirements to the system and the preconditions of the environment in mind, we will illustrate the necessary design decisions of our Internet voting system in the next chapter.

4 Design

As illustrated in the introduction there is a variety of alternative solutions to design an Internet voting system. However, not all of them will meet the requirements given in chapter 3. Some of the design decisions are indispensable:

Polling Station vs. Individual Computer Internet voting must take place at the polling station at present because the use of individual computers is not conformable with the requirement that everybody can vote regardless of her personal havings and it also violates the assumption that only trusted secure platforms must be used. We cannot guarantee the absence of Trojan horses on personal computers which might corrupt the secrecy and integrity of the overall system.

Authentication A next design decision concerns the issue of authentication. The use of digital signature cards combined with personal identification numbers (PIN) currently is the best compromise between security and minimizing the resulting costs of implementing the technology (compared for instance with using personal fingerprints). Using qualified signatures, as described for instance by the German Digital Signature Act, the requirements for authentication can be satisfied. This aspect implies another design decision: it is essential to establish a certificate authority that creates the certificates to check the validity of the voters signatures.

Divison of Power Each voting system must respect the principle of the division of power because otherwise (as we assumed in the definition of our trust model) an attacker would be able to corrupt the system by manipulating the single component. It is important to notice that the division of power enforces the separation of computations in the following three situations: Two components are needed for authorization check. A single component would permit unauthorized people to vote or to exclude authorized voters from voting, for instance, by changing the electoral register. This would contradict the requirement that an attacker is not successful if he manipulates only a single component.

The second situation occurs within the polling booth. Because we require that votings are kept secret and assume that an attacker can manipulate a single component, we also need two components in the polling booth. One component is concerned with the registration and the processing of voter's information and the other component is casting the votes without knowing anything about the actual voter. Even if one of these components is attacked, there is no allocation from the voter to her decision possible. Finally, it is essential to separate ballot collection from result calculation to prevent the calculation of intermediate results. This means that there is a component which simply collect all ballots but which is not able to calculate intermediate result. After reaching the election deadline all ballots are transferred from this component to a second one which will calculate the result of the election.

Beyond Cryptographical Secrecy There are two additional design aspects from the given legal requirements: The first aspect is concerned with the electoral secrecy which must be guaranteed also in the future. It is hard to predict how progress in computer hardware and cryptography will damage probabilistic properties of existing cryptographic approaches. Additionally, we assume that the attacker is able to read all transmitted ballots and he can observe who actually is in the polling booth at a given point in time. Therefore it is not sufficient to use encryption - neither asymmetric nor symmetric - if the component transmits the ballot immediately. An attacker will know the allocation between voter and her decision as soon as the underlying cryptographic approach is broken. A new mechanism similar to MIXEs [Cha81] is needed to conceal the relation between a voter standing in the booth and the votes being sent from one component to another. We will discuss the details of our mechanism in the following section. However, even if we use such a mechanism, the encryption of ballots is still essential for another reason: to prevent intermediate results, which must be confidential until the end of the election (This encryption is the second design aspect).

Summing up, the architecture of the proposed Internet voting system consists of two components which check the authorization, one component to collect the votes and another one to compute the result. Furthermore, there are two components in each polling booth. One component is concerned with the authorization of the voter while the other component is used for the actual voting.

5 Realization

Based on the analysis presented above, we developed an Internet voting system called *SecVote*. In this section we will describe the architecture of the system (cf. Figure 1) which consists of the following six components:

The **Registration Server** (RegServer) and the **Certificate Authority**[1] (CA) that are responsible for authorization check, the **Voting Box Server** (BoxServer) that collects the votes and stores the content of all ballots, and the **Control Server** (Controller) that computes the final result. The **Registration PC** (RegPC) that deals with the authentication for access and the **Voting PC** (VotePC) to cast the voter's ballot (both in the polling booth).

Protocol The protocol (cf. Figure 1) of the voting process works as follows: The voter enters the polling booth and is informed by the RegPC to activate her signature card using her individual PIN code.

[1] The Certificate Authority is used for two tasks: first to check the cert validity and second for the authorization of voters.

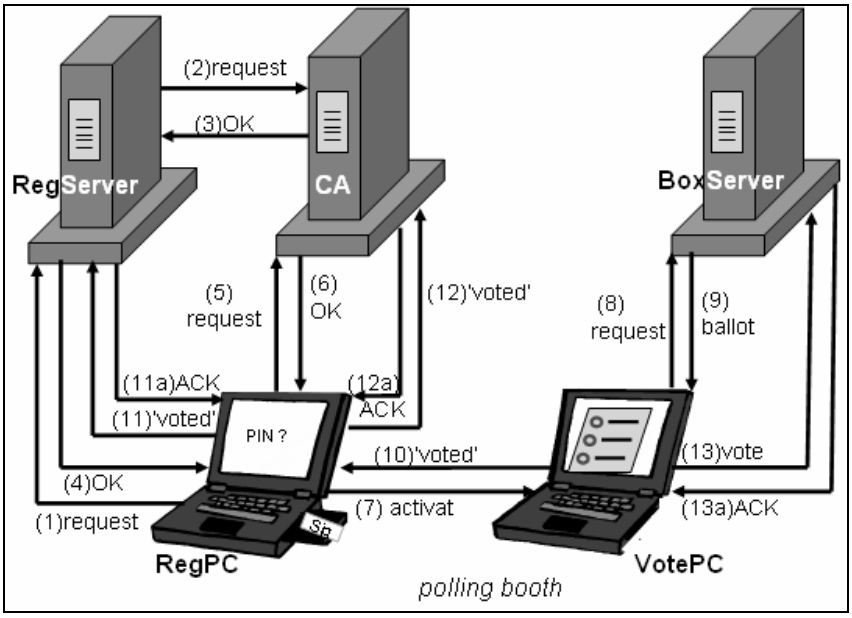

Figure 1: Architecture and Communication

The RegPC sends a request both, to the RegServer (1) and to the CA (5). Receiving the query, the RegServer checks the voting authorization and sends a validity request to the CA (2). The CA, getting the message, checks it against its revocation list to see whether the cert is still valid, and sends the answer back to the RegServer (3). The RegServer forwards this answer to the RegPC (4). In addition the CA receives a request directly from the RegPC (5). Before sending the answer to the RegPC (6) it checks the voting authorization and the cert validity. If the RegPC receives the acknowledge from both components, RegServer and CA, it sends a message to the VotePC (7) to activate the voting process and informs the voter that she should proceed to the second PC. This PC first asks the BoxServer for the content of the ballot (8) and displays it to the voter after receiving this information (9). Next the voter has to make her decision and to acknowledge it. Then, the VotePC informs the RegPC (10) to change the status of the actual voter in the election register and sends the ballot to the BoxServer (13). The RegPC forwards the information about the end of the actual voting to the RegServer (11) and the CA (12). Both components adjust their internal database and send acknowledgments to the RegPC (11a, 12a). The BoxServer stores the ballot and acknowledges it (13a). Both, VotePC and RegPC display a message that the ballot was casted successfully and that the voter can remove her signature card. The system is now ready to welcome the next voter in the polling booth.

The sketched design of the system (architecture and protocol) is not sufficient to ensure the given overall requirements. Additional mechanisms are needed to meet these requirements. Some of them are obvious: e.g. all messages have to be digitally signed to

obtain integrity and authenticity. A back-up-system is required to safeguard the availability of the system, access control mechanisms are necessary to gurantee the privacy and integrity of data on individual hosts, and mechanisms are needed to ensure secure data transfer.

Secrecy of election and uniqueness of ballots This section will illustrate the mechanisms used in *SecVote* to keep the **election secret** and to prevent that ballots are deleted, changed or added. The main problem with the secrecy of elections is the assumption that eventually in the future an attacker will be able to decode the recorded encrypted votes sent from the VotePC to the BoxServer. Although the votes do not contain any information about the voter, the attacker might still be able to monitor the polling station and relate the physical presence of a voter in the polling station with the shortly following message of the VotePC to the BoxServer.

Therefore, we use a similar approach to MIXEs [Cha81]. The VotePC does not immediately transmit the voter's ballot but the first casted ballot is only stored within the VotePC. Two ballots always remain in the memory until the next person casts her vote. The VotePC transmits now one of these two to the BoxServer. The choice is absolutely random. Thus an attacker does not know whether the transmitted ballot correspond to the first or to the second voter. He can only make a guess with a probability of 0.5. The same procedure takes place for the following voter and all others. After finishing the election the VotePC sends the last stored ballot to the BoxServer. This ballot can be either from the first, the last or any other voter. Hence the attacker, once able to crack the cryptography, only knows that either the last or the last but one transmitted vote belongs to the last voter in the polling station.

There is one case in which the attacker will know the decision of the last voter in the election once he is able to decode the encrypted messages: If the last and the last but one transmitted ballot are equal then the attacker is able to allocate this decision to the last voter of the election. However, on the one hand the probability of this event is very small[2] and the attacker cannot precipitate such a situation. On the other hand the attacker only knows about the decision of a randomly affected voter but cannot use this weakness to get hold of the decision of a previously selected person. So this fact does not affect the trust model and the proposed procedure can be used to safeguard the secrecy of the election.

Within *SecVote* we have incorporated three mechanisms to **ensure the correctness of the voting result**: To prevent that ballots are copied or modified, all messages are signed together with a unique random number. The Controller verifies all signatures and checks that all numbers are unique. Apart from that, the Controller compares also the number of received ballots with the number of voters in the election register from the CA and the RegServer. Thus, any deletion of votes will be revealed. To ensure that the VotePC transmits or stores the correct ballot, the signature is generated on an external secure signing component (Signierkomponente) equipped with a separate screen.

---

[2] The probability depends on the number of possible votes and becomes exponential smaller if you collect more than two votes before sending once.

6 Implementation

SecVote was implemented as a proof of concept of the presented design of an Internet voting system. It includes most of the functionality outlined in this paper and was implemented in a collaboration between the Federal Office for Security in Information Technology (Bundesamt für Sicherheit in der Informationstechnik) and the German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz).

Its main parts are implemented in Java. The used cryptographic algorithms are RSA [RSA78] with SHA-1 [NIS92] for digital signatures, IDEA [Lay92] for symmetric encryption and a pseudo random number generator from Sun - however for a legal election it must be replaced with a perfect random number generator.

7 Related Works

There is a vast number of literature concerning Internet voting, the development of systems and the test of resulting systems. The published work can be divided into work on Internet voting (including suitable protocols for communication) allowing voters to use their individual personal computers and work on voting based on polling stations.

Examples for individual Internet voting are described in [Sch00] and [Cha81]. However, this class of voting systems, which will run on non-trusted hardware, does not conform with the legal standards presented before. The emphasis of most of these papers was put on two requirements: to ensure the secrecy and the integrity of the election. They abstract from the unsolved problem of voting using untrusted hardware and operating systems and the problem of ensuring that all voters are equipped with the necessary systems. However, without solving these problems the use of these proposed systems would lead to a violation of the principle of universal suffrage.

The other group of papers is addressing the problems of individual platforms and propose the use of polling stations for voting systems. Most of these voting systems, like for instance [FOO93], [PKKU02] and [BY86], adopt the principle of the division of power. These voting systems fulfill at least some of the mentioned design decisions. But they do not unconditionally ensure the election secrecy. They use, for instance, only encryption to ensure the secrecy of ballot transmission (e.g. i-vote [IVO02] uses RSA) but neglect the fact that any used encryption mechanism based on probabilistic results might be cracked in the future. It is insufficient only to separate votes from information about the voters. This could result in a violation of the legal principles in the future.

Besides the design of these systems there are additional problems arising with the implementation of such existing Internet voting systems. To ensure economical success, developers of these systems do not publish detailed information about the system and do not speak about the source code. Since these systems are also not certified by a trusted third party, voters will have to trust in the developers that everything works correctly. But this lack of control results that most voters will not accept such systems.

8 Conclusion

In this paper we illustrated how to develop an Internet voting system for legal and binding elections. This proposed system is in accordance with German laws, which are very close to those in other European countries. The described design, following the principle of division of power for the design of the architecture and inventing a random-mechanism for transmitting ballots, ensures legal standards and especially the unconditional secrecy of the election regardless of future developments in cryptography. Furthermore our system is robust in a sense that it will notice forgeries even if the attacker is able to manipulate a single component.

Literaturverzeichnis

[BY86] Benaloh, J. C.; Yung, M.: Distributing the Power of a Government to Enhance the Privacy of Voters; In: Proc. 5th Symposium on Principles of Distributed Computing (New York, USA: ACM 1986), pages 52-62,1986.

[Cha81] Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, University of California, Berkeley -Communications of the ACM, 24: 84-88, 1981.

[FGr] Internetseiten der Forschungsgruppe Internetwahlen mit Informationen zur Software und zu den durchgeführten Projekten; www.internetwahlen.de.

[FOO93] Fujioka, A.; Okamoto, T.; Ohta, K.: A Practical Secret Voting Scheme for Large Scale Elections; In Advances in Cryptology - AUSCRYPT 93; Springer-Verlag; pages 244-251; 1993.

[ISO00] ISO/IEC International Standard; Common Criteria for Information Technology Security; Evaluation (CC); Version 2.1; ISO IS 15408; csrc.ncsl.nist.gov/nistpubs/cc/; 2000.

[IVO02] Abschlussbericht zur Online-Wahl im Landesbetrieb für Datenverarbeitung und Statistik im Land Brandenburg; www.forschungsprojekt-wien.de/pdf/lds.pdf; page 23; 2002.

[Lay92] Lay, X.: On the design and security of block cipher; In ETH Series in Information Processing; 1992.

[NIS92] NIST: Proposed Federal Information Processing Standard for Secure Hash Standard – FIPS; National Institute of Standards and Technology (NIST); 1992.

[PKKU02] Prosser, A.; Kofler, R.; Krimmer, R.; Unger, M. K.: e-Voting.at: Entwicklung eines Internetbasierten Wahlsystems für öffenliche Wahlen; Arbeitspapiere zum Tätigkeitsfeld Informationsverarbeitung und Informationswirtschaft; 2002.

[RSA78] Rivest, R.;Schamir, A.;Adleman,L. M.: A Method for Obtaining Digital Signature and Public-Key Cryptostreams; In Communications of the ACM; 1978.

[Sch96] Schneier, B.; Applied Cryptography; John Wiley & Sons; 1996;

[Sch00] Schoenmakers, B.: Fully Auditable Electronic Secret-Ballot Elections; 2000.

[Wil02] Will, M.: Internetwahlen - Verfassungsrechtliche Möglichkeiten und Grenzen; LL.M. (Cambr.), Institut für Öffentliches Recht Philipps- Universität Marburg; Richard Boorberger Verlag GmbH & Co; Recht und neue Medien Band 2; 2002.