Technical Report

Nr. TUD-CS-2011-2871
January 16th, 2011

CASED

Authentication Techniques, Client-Side Voting Software, and Secure Platform Mechanisms

Dimensions of Remote Internet Voting

Authors
M. Maina Olembo and Melanie Volkamer

Authentication Techniques, Client-Side Voting Software, and Secure Platform Mechanisms

Dimensions of Remote Internet Voting

- **Abstract:** *Electronic voting is still a hot topic. You can mainly distinguish between three types of electronic voting, namely direct recording electronic voting devices in polling stations, scan based electronic voting systems, and remote electronic voting. In this paper we focus on the last category and in particular we discuss three dimensions that are specific for remote Internet voting. These are voter identification and authentication techniques, client-side voting software used to cast the vote and secure platform mechanisms to overcome vulnerabilities of the client used by the voter to cast her vote. We describe and analyze different implementations of each of these in the context of remote Internet voting, and assess their performance based on usability, security, costs, and maintenance issues. We identify combinations that cannot be applied at the same time and make recommendations for the application of particular implementations for specific types of elections.*

Keywords: Authentication, client-side voting software, secure platform problem, remote Internet voting

1. Introduction

Internet voting, website voting and online voting are terms used interchangeably to refer to situations where a voter can cast her vote using the Internet as a transmission medium. Within this context, according to Oppliger (2002), there are different implementations that can be provided such as poll-site Internet voting, kiosk Internet voting and remote Internet voting[1]. Remote Internet voting, allows a voter to cast her vote from any location, whether from home or work on the selected election day(s). It is just required to have a device (in general a PDA, PC, or a laptop) that is connected to the Internet. This presents several benefits: It enables the voter to vote from a location and at a time of her choice, thus enhancing convenience and flexibility for the voter. Thus it could potentially increase voter turnout. Remote Internet voting, like any other type of electronic voting makes the tallying process faster and verification of results simpler for the electoral body, as well as leading to reduced costs, particularly in the long-term.

There are several challenges in remote Internet voting which do not exist in other types of electronic voting. Consider that it is carried out in an uncontrolled environment not directly under the supervision of the electoral body or its officials and without a physical voting booth. In this scenario, the voter needs to be authenticated as an eligible voter over the Internet as no poll workers are around to check the voter's ID card against the electoral register. Furthermore, the voter or a third party controls the environment including the place to cast the vote, the device to run the voting software and the software to communicate with the voting servers. Regarding the voting software, the voter might use a malicious one – either on purpose as she wants to sell her vote or without her knowledge. In addition, there exists the risk of malicious software such as viruses and Trojans on the voting device. The payload of both attacks is either to change the voter's selection or to break the secrecy of the vote. In order to overcome these problems a lot of different solutions have been proposed. However, what is currently missing to our knowledge is a comprehensive list of the different solutions including a description of each approach and an analysis of the advantages and

[1] Another type of remote electronic voting is SMS voting which is not addressed in this paper.

disadvantages combined with a comparison. In addition, statements regarding possible combinations and impossible ones are missing. As all this would be of great help for election bodies, we will fill this gap with this paper. Our comparison will focus on security and usability aspects as well as on additional costs and additional effort for maintenance.

The paper is structured in the following way: In Sections 2 we discuss different authentication techniques in the context of remote Internet voting, the advantages and disadvantages of each implementation, and identify those approaches that are obviously not applicable for any type of election. Afterwards, in Section 3, different approaches for the client-side voting software and in Section 4, different mechanisms to overcome the secure platform problem are introduced and discussed. In each section we present an analysis of existing approaches where we compare the different approaches regarding security and usability requirements as well as the costs and the effort for maintenance. Finally, in Section 5, we summarize the paper, present our findings, and make recommendations for future work.

2. Authentication Techniques

The process of authentication aims to ensure that a voter is who they claim to be by verifying the stated identity. Every remote electronic voting system needs to implement identification and authentication techniques to ensure that only eligible voters can cast a vote and those only once. Thus, the voter needs to be electronically identified and authenticated. The identification must be unambiguous. This can be solved by name, surname, birthday, and/or social number or other unique numbers. Therefore, we only focus on the authentication technique in this paper. In information security, three types of authentication forms are known as well as a combination of the three: something you know (a password), something you have (a token), and something you are (fingerprint). In the following subsections, they are described and discussed in the context of remote Internet voting.

a. Something You Know: a Secret

The first category of authentication techniques is based on knowledge. There are two possible implementations while each can be applied in different ways: The first possible implementation of voter identification and authentication is applied in accordance with the setup of an e-mail account: in the election setup phase, voters create their voting accounts, which will later be used by the voters to cast their vote. Different from the e-mail account where – in general - everyone can create an email account on behalf of someone else, the voter needs to be authenticated beforehand as an eligible voter, e.g. at some central offices, and create the account afterwards at this central office. Otherwise it cannot be excluded that other persons who are not eligible for this particular election can set up an account. Correspondingly, this solution requires some additional effort from the voter as she needs to go to these central offices in the election setup phase. There are two possible settings; depending on the frequency of elections, voters might create a new account per election or can use the same account for several elections.

Regarding vote casting, this approach is easy from the voter’s perspective as the voter is used to login/password authentication from many other Internet applications. However, it also has disadvantages: First of all, voters might choose weak passwords which can be easily hacked by an attacker or they might write their passwords down. Thus, friends or family members can get access to their password. Second, vote buying is not excluded because voters could send their login data electronically to a potential buyer. There is nearly no effort for the voter and it is free of charge to send an email to the potential buyer with this information. How realistic it is that voters forward their password to a potential buyer depends on whether the same account can be used for several elections or not. Probably, even if the same account is used for several elections the voter could ask to reset the password and, thus, the vote buyer from the last election cannot use the password she got from the last election for the next one. In addition, the probability that someone will pay for the account details is very low if vote updating is enabled. The problem for the buyer in this case is that she does not know whether the voter does not also cast a vote some seconds before the end of the election phase and thus overwrites her ‘bought’ vote.

Another approach to implement authentication through knowledge of a secret is based on a so called voting TAN procedure. A voting TAN is a transaction number, similar to those used for online banking. The vote TAN is a unique code of several (more than four) letters and digits for each voter and is sent by post to eligible voters in the election setup phase. In general, the voter will get a new voting TAN for each election. Usually, the voting TAN is covered by a scratch field. Thus, if for example, the secretary opens the post for the boss, she does not get access to the voting TAN without being detected (by the destroyed scratch field). This variation is rather similar to the one above with respect to usability issues. The advantage is that the voter does not need to go to such a central office. Security increases as the voting TAN can be generated through the responsible election authority as a strong secret. However, the costs increase since the eligible voters get their voting TAN by mail. To ensure that no voter looses the right to vote you need either to trust the postal service and the printing service of the voting TANs or you need to implement procedures that enable voters to cast a paper vote in case they claim not to have received the voting TAN. Furthermore, you need to have a procedure to check that these voters do not cast two votes and regulations how to react if two votes have been cast. The risk that the TAN will be forwarded to an intruder in order to sell the vote still exists and depends on whether vote updating is enabled or not. For both these approaches there are no special maintenance requirements.

b. Something You Have: a Token

The second category of authentication techniques is based on ownership. Two different implementation possibilities can be identified for remote Internet voting: In one implementation, a new election specific authentication card is used, which will be sent to the voter prior to the election (similar to the TAN from the above category). Again a new card could be sent before each election or there is one card that is used for all later elections, too. Compared to the TAN solution this one provides more security since the buying of votes is more expensive because getting the card is more difficult than getting a copy of the TAN, for example, via email. However, as the card has no other purpose than casting a vote, people who do not care about the outcome of the election might forward the card to potential buyers. Different from the last approach, here, it does not help to enable vote updating. The costs for the election bodies rise substantially: apart from production and distribution costs of the smart cards, substantial costs of an appropriate card reader arise either for the voter or for the election bodies, too. From a usability perspective, negative effects will probably appear caused by the necessary installation of the card reader and corresponding software on the voter’s PC.

In the second implementation, a pre-existing authentication card is used, which the voter already owns and uses for authentication purposes in other areas, like her ID card, job card or library card. Thus, someone who has this card obtains access to other services too and is not just enabled to cast a vote on behalf of the card owner. Correspondingly, such a card will not be lightly passed on to a vote buyer, since this automatically means that all other applications of the card are passed on as well. Additionally, the use of an already-owned card increases the user-friendliness (at least a bit) as the voter might also be used to authenticating himself with this card. However, the costs of the card reader remain if the voter does not yet possess such a device. From our perspective, the user-friendliness will not really be increased as such cards are usually used for authentication at terminals (to enter rooms, borrow books, pay lunch in the canteen, and so on) and not at own devices. Therefore, the installation effort remains as well as the costs to buy and distribute the readers. Regarding maintenance, the effort increases for both solutions compared to the implementation of authentication techniques based on secrets as there is software, drivers, and a device at the voter's place which need to be maintained, in addition to the components on the server side.

c. Something You Are: Biometrics

The third authentication category is based on biometric attributes[2]. Examples of biometric attributes are fingerprints, iris scans, face recognition (size and position of different facial features), voice (mode and tone while speaking), manual signature (form and dynamic aspects), and DNA. The form or structure of each of these attributes is unique per person. In order to authenticate a person the corresponding attribute is scanned. The scanned copy of the attribute is then compared to the one stored for the subject. In case it matches, the subject is authorized, otherwise she is rejected. The main advantage of biometric authentication is that attributes cannot be forwarded to another person, for instance, vote buyers. Therefore, from a security point of view, it does not need to distinguish between systems that are already deployed and those that are introduced for the election. However, from the cost and user-friendliness point of view it would make a difference. Unfortunately, the matching of scanned and stored data does not work perfectly: the system can falsely reject an authorized subject, or it can falsely accept an unauthorized subject. Therefore, each system has a False Rejection Rate (FRR) and a False Acceptance Rate (FAR). In the past, the FRR has been disregarded as FAR is much more important for privacy and integrity issues. In elections, availability is (because of the universal election principle) as important as other properties. In addition, a major point of concern for a biometric system is how to securely store such sensitive data and due to data protection requirements it is not recommended to store biometric data from all voters in a central database. However, there is no other solution if you do not want to combine this approach with other authentication techniques. Another disadvantage involves the costs for introducing such an approach for elections because large-scale biometric infrastructures in general do not exist, yet. Furthermore, every voter would need a reader at her place, corresponding software and drivers. Regarding usability issues, this of course has the consequence that people need to be able to install the corresponding software and drivers.

d. Combinations of Different Techniques

Often, a combination of the above listed authentication techniques is used. The most popular ones are the combination ownership/secrecy in signature cards, especially if used for qualified digital signatures, and ownership/property to store the sensitive biometric data on a smart card instead of in a central database. The idea would be similar in both cases. The smart card needs to be enabled before it can be used for authentication purposes. This is done either by a PIN code or by the scanned biometric data. The application of these combinations maximizes the security because in both cases it is hard to fake the card. In the case of the biometric properties, a vote buyer cannot use a voter's card because she cannot enable the card without the biometric data from the voter. In the other case this risk in general exists. However, if such a card is used for several applications, forwarding the card means giving someone else the opportunities to get access to other data and applications or to legally sign any document. Neither of these two implementations is widely distributed. There are contexts and countries where the combination of smart card and PIN is widely spread as the ID card is such a card. However, even in these contexts, the corresponding functionality is not used as users have no corresponding card readers at home. This means that there might be no costs to develop, build, and distribute such cards but still the card readers need to be bought, distributed, and installed. Regarding maintenance, the additional effort is similar to simply using cards and a reader for authentication.

[2] Biometrics and their applicability to electronic voting is discussed in detail in (Hof, 2004)

e. Comparison

The above discussion shows that there does not exist ‘the’ best solution if you analyze the approaches according to the aspects usability, costs, security, and maintenance effort. There are solutions that are more likely to apply than others however the decision for which solution to go for should also depend on the environment in which the system will be used in (e.g. is there already a public key infrastructure or other authentication techniques deployed) and the type and importance of the election as the risk of a manipulation for an election in societies is different from the one for federal elections). An overview of the above analysis is displayed in figure 1 (level 5 is the optimum one can reach).

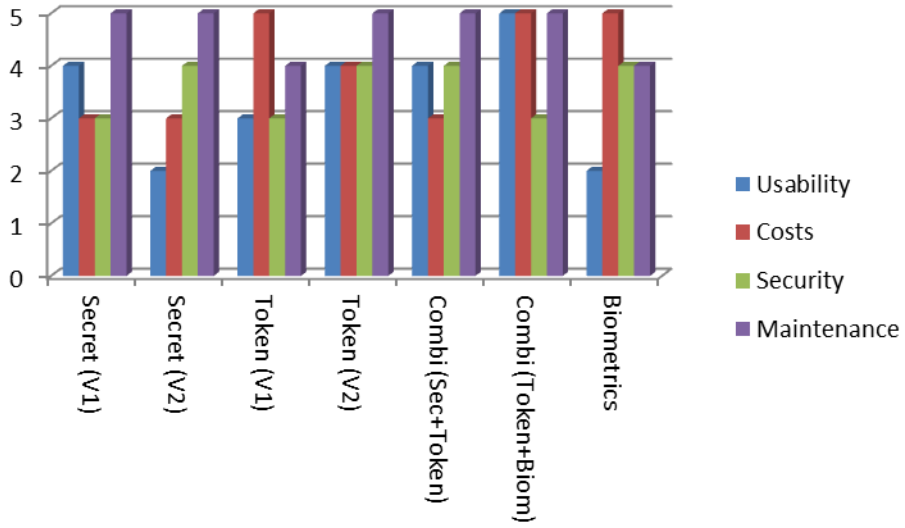

Figure 1: Authentication techniques and their applicability for remote Internet voting

3. Client-Side Voting Software

The client-side voting software is essential for the voter to communicate with the voting server. The client-side voting software runs on the vote-casting device. It can either be a web browser, a specific application – a so called rich client, or a particular voting applet – a so called thin client which can be run in the Web browsers. These three different approaches are proposed and discussed in the following subsections.

a. Web Browser Solution

One possibility to enable the voter to communicate with the voting server is the application of available Web browsers without any specific client-side voting software. This approach does not involve any kind of java applet. Due to the poor security functionalities of a Web browser, the main security mechanisms of the voting system run on the voting server. The Web browser is only used to establish the link to the voting server, to display the voting Web page, and for the voter to interact with the voting server including authentication of the voter and vote casting. The only assumed security functionality is the Secure Socket Layer (SSL). This is necessary to secure the communication because Web browsers do not provide another possibility to encrypt or sign messages. Using SSL, it is possible to ensure confidentiality and integrity of the exchanged messages. Moreover, the authenticity of the voting server can be ensured. Note, this can only be ensured with the help of the voter who needs to check the voting server’s certificate.

From a usability perspective, the Web browser solution is welcomed because the voter does not need to install additional software but can use the environment she is used to. Moreover, Web browsers are executable on other devices, other than normal PCs or notebooks. The voter can also use her WAP mobile phone or PDA to cast her vote using the remote Internet voting system. The only disadvantage from the usability point of view is the necessity for the voter to check the certificate of the voting server. This might be new for many voters even if they use SSL on a daily basis. To improve the situation it might help to use so called extended validation certificates. Concentrating the whole functionality on the server-side has two more advantages: first of all, in the case of a voting system update there is no effort for the voter because only the server-side voting software needs to be updated. Secondly, if a new Web browser or a new version of an existing Web browser is deployed the server-side voting software can be patched in order to support this new Web browser as well. Thus, the voter is free to choose the Web browser she prefers[3] to cast her vote.

However, there are two main disadvantages: first, the remote Internet voting system has no possibility to check the trustworthiness of the vote-casting device, for example, whether there is a virus or Trojan horse on the vote-casting device which affects the communication between the voter and the voting server. Moreover, an (un-patched) Web browser could weaken the trustworthiness by well-known exploits. The second disadvantage is caused by poor Web browser functionality. Thus, most of the proposed voting protocols cannot be implemented because this would require security functionality on the client-side. For the same reason, this approach can only be used in combination with secrets as authentication techniques.

b. Rich-Client Solution

This second approach is called rich-client because the client-side voting software is rich with respect to security functionality including the implementation of different cryptographic algorithms. This client-side voting software needs to be installed and executed on the vote-casting device in order to cast a vote. Any available voting protocol can be implemented using the rich-client approach, thus, in contrast to the Web browser solution; this solution does not exclude any voting protocol or any authentication technique. In addition, a rich-client can include a virus scanner or similar security software in order to verify the trustworthiness of the vote-casting device before starting the vote casting process as proposed in (Jones, 2002). Note, the voter needs to agree and may also not want voting software searching her file system for viruses or the like (for fears about privacy or the federal Trojan horse).

The disadvantages of this approach are the distribution, installation, and maintenance of the client-side voting software. Distribution and maintenance is an economic question while the installation is assigned to usability issues. Moreover, the client-side voting software might only run on a particular system if corresponding system properties are given (for example, the java virtual machine is running). Analogous to the validation of the server certificate in the Web browser approach, the voter needs to verify the integrity and authenticity of the voting software she installed on her vote-casting device. This is even more complicated for voters than the verification of a server certificate.

c. Thin-Client Solution

The Web browser solution is preferable from a usability and maintenance aspect whereas the rich-client is advantageous from a flexibility and security point of view. A mix of both strong points is provided by the thin-client approach. In this approach, a java applet is implemented to run in the Web browser. This java applet is the client-side voting software which provides the necessary security functionality on the client-side. The thin client approach offers several advantages. Any authentication technique and voting protocol can be implemented. Maintenance is easier since only the server-side software needs to be updated. It is easier to support new web browsers and new versions of web browsers. Furthermore it is easy to use, no software installation is required, it can be used with other

[3] Even more, the voter needs to be free to choose a Web browser she wants to use; a particular one cannot be mandated by the responsible election authority

voting devices and the voter is free to use a web browser of their liking. However, Java Script has to be enabled.

d. Comparison

Similar to the discussion of the dimension ‘authentication technique’, the above discussion shows that there exists no best solution for the communication between voters and the voting server if you analyze the existing approaches according to the aspects usability, costs, security, and maintenance effort. All solutions have been applied for real remote Internet voting while the solution that was most likely in use is the thin client approach. Here you should have in mind for your decision that even the web browser solution provides the best usability you cannot combine it with either existing cryptographic voting protocols or any authentication techniques other than secret based. An overview of the above analysis is displayed in Figure 2 (level 5 is the optimum one can reach).

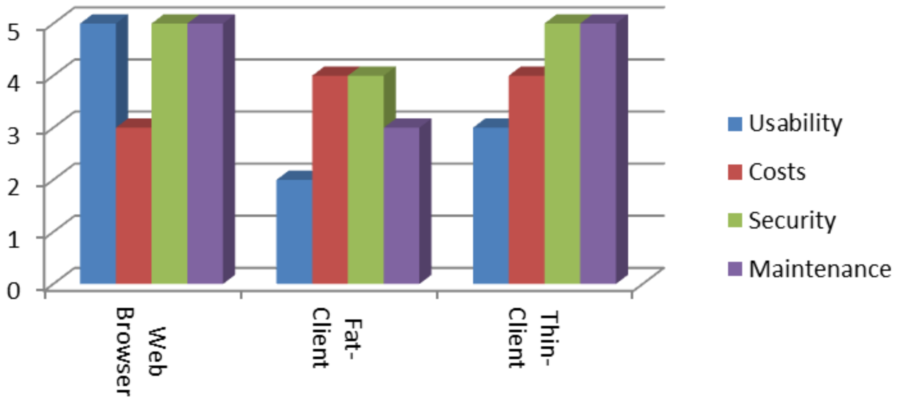

Figure 2: Approaches for the client-side and their applicability for elections

4. Secure Platform Mechanisms

The Internet is considered an insecure environment. As mentioned, running elections over the Internet presents several challenges including the possibility of malicious software on the vote casting device modifying a voter's vote without their knowledge or breaking the secrecy of the vote. In order to run an election one needs to overcome this so called secure platform problem. In this section we discuss and analyze several approaches which have been proposed to overcome this problem.

a. Guidelines

A first approach to address the secure platform problem is the provision of special guidelines explaining to voters how they can improve the trustworthiness of their vote-casting device. Examples are the guidelines provided by the Swiss government for the remote Internet project and those developed by the German society of computer scientists (GI, 2010). This approach can reduce the risks created by malware on the vote-casting device. However, you probably mainly overcome standard and well known attacks. In addition, one cannot force voters to apply the security checks from the guidelines and many voters are not likely to be able to follow the instructions. Moreover, such an approach is useless against malicious voters installing malware on purpose. Regarding usability, one can say that the probability is very high that voters do not follow the instructions because these are additional steps or they are simply not able to do so (depending on which instructions are included in the guidelines). The costs are not very high as you only need to develop a list of instructions and then send it via email or put it on the election web page. Similarly, the maintenance effort is low as only the ‘piece of paper’ needs to be updated.

b. Bootable CD

Another approach to overcome the secure platform problem was proposed by Otten (2005). She recommended developing a special voting operating system based on Knoppix. Knoppix is an operating system based on Debian that is designed to be booted and run directly from a CD or DVD. Such a CD would then be distributed to the voters. After having received the CD, voters need to configure their vote casting device in a way that it boots from this CD. Probably, additional security checks are required from the voter to overcome the risk to have received a malicious CD which communicates with a malicious server with the consequence that the vote can be altered before it is cast to the proper voting server and the secrecy of the vote can be violated. A main challenge of this approach is to develop a CD that boots from all the different hardware around and can be automatically connected to the Internet with all the different providers available. Probably the voter needs to take additional steps to configure the Internet connection. Note, even if all this could be solved this approach does not solve the malicious voter problem as it only prevents attacks caused by malware. Voters could for example install additional malware on top of the system booted from CD. The answer to the question depends on the quality of the CD. If the voter needs to do a lot of configurations, the approach is not very user-friendly but if the operation system does everything without the voters help it would be a user-friendly approach. Regarding cost, there are high development costs and the CDs need to be distributed to the voter in a secure way. This CD needs to be updated for the next election or for one of the next elections (depending on the time frame between two elections) and a new CD needs to be sent to the voters. Correspondingly, both the voting server software and the bootable CD need to be maintained.

c. Smart Cards as Observers

An observer is a manipulation resistant piece of hardware which is owned by the voter. The idea is that the observer is not allowed to directly communicate with the Internet. All the communication needs to be forwarded by the voter. The concept of an ‘observer’ has been introduced in (Chaum & Pederson, 1992) and has been refined in (Cramer & Pederson, 1993). It has been further developed and extended by Schweisgut (2006) and Juels et al. (2005). Here they proposed the application of a smart card as an observer. By applying such an approach, one overcomes most of the attacks from malicious voters. However, a smart card does not interact directly with the voting server but over the vote-casting device. Malware on this device can mount a man-in-the-middle attack and misuse the card, for instance, by sending a wrong candidate choice to the smart card or the vote-casting device displays a modified ballot. Regarding usability, there are the issues to get such a card and a corresponding reader and install and configure the reader. The costs are higher than for existing smart cards as it needs to ensure special properties. Further, it needs to be distributed to all voters. Regarding maintenance, one has one more component, the smartcard and maybe also the reader to maintain.

d. Code Sheets

The idea of code sheets is that the voter gets a piece of paper together with the general election information via post where each candidate or each party is linked to a particular code. Now, in order to cast a voter the voter does not click on the candidate or party of her choice but enters the corresponding code. This idea was first proposed by Chaum (2001). Helbach & Schwenk (2007) propose to use the code sheets to overcome the secure platform problem and Oppliger et al. (2008) proposed an improvement of this technique. This code sheet is sent via ordinary mail and contains for each candidate a voting TAN (the origin code) and a confirmation TAN. To overcome vote selling the authors introduced in (Oppliger et al., 2008) an additional TAN – the so called finalization TAN. The voter enters a corresponding voting TAN instead of choosing a candidate on the PC screen. To verify the correctness, she compares the received and displayed confirmation TAN with the one on the code sheet. In the case of (Oppliger et al., 2008), the voter would confirm the correctness with a third TAN. One disadvantage of this approach concerns the user-friendliness which decreases in particular for implementing complex ballots. In addition, a trusted procedure to generate and distribute the codes/TANs is required. The costs are relatively low as election information letters are sent anyway and one can add this code sheet to this letter. Additional requirements for maintenance do not occur. Note, this approach works best if vote updating is enabled.

e. Trusted Computing

Another approach proposed to overcome the secure platform problem is based on trusted computing techniques. In particular, the idea is to use an appropriate security architecture based on a security kernel and on Trusted Computing elements. Such a solution is the only one that could efficiently overcome malicious software on the voting casting device as well as potential malicious voters installing malware on purpose on their device. However, currently, there are still open problems with Trusted Computing itself and it is not wide-spread enough. People might have a laptop with integrated Trusted Platform Module (TPM) but the security architecture and security kernel are missing. A more detailed discussion of this approach is provided in (Alkassar et al., 2006) and (Volkamer et al., 2006). This approach provides high security but as it is not yet available one needs to conduct further research on it for future elections. If this approach is done properly, there will also be no usability concerns. Maintenance can also be solved. However, there are high development costs before it can be used. Once the techniques are available, additional cost per elections are relatively small.

f. Individual Verifiability

As another approach to overcome the secure platform problem, we want to mention individual verifiability mechanisms. The idea is that you use one software to prepare a voter and a second one to verify that the vote has been properly prepared (encrypted). Plus, you can also do the verification with an offline tool. Now, you can still develop malware to alter voters undetected but it requires more effort as both software tools need to be manipulated. In addition, an attacker does not know whether the voter might go for offline verification. In this case, the manipulation would be detected. Furthermore, such a mechanism does not help against malware with the objective to break the secrecy of the vote. Therefore, this mechanism should be combined with vote-updating. Regarding usability, one must say that there exists almost no research on the usability of verifiability. First papers like (Weber & Hengartner 2009) show that it is a huge challenge to provide user-friendly verifiable remote Internet voting systems. However, having a verifiable remote Internet voting system you automatically get a mechanism to overcome the secure platform problem without any additional costs. Correspondingly, there is no additional effort for maintaining a particular mechanism to overcome the secure platform problem. Note that this approach does not prevent voters from installing malware on purpose.

g. Comparison

Similar to the other two dimensions, the secure platform was found to offer a variety of results with respect to usability, security, costs, and maintenance. It was noted that again no best solution exists. E.g. the security architecture (trusted computing) scored low in usability and yet the costs, security, and maintenance were high. This reiterates again the well-known fact that there is normally a tradeoff between security and usability. Here one should take the environment into account. For instance, if you allow vote casting from Internet cafes you have a much more insecure environment than when you have an election in a company with centralized administration and correspondingly much more secure platforms. An overview of the analysis is given in Figure 3 (level 5 is the optimum one can reach).

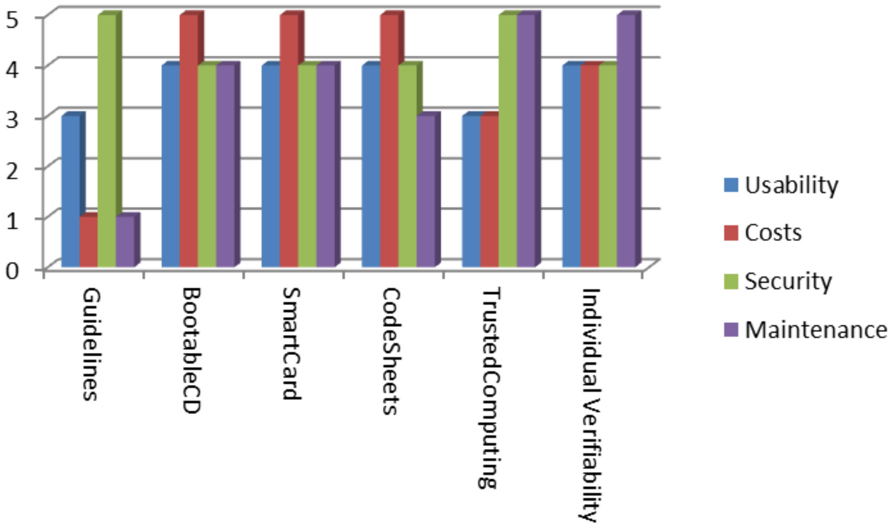

Figure 3: Trusted Platforms and their applicability for elections

5. Conclusion

In this paper we have focused on special challenges of remote Internet voting and in particular on three dimensions of remote Internet voting, namely authentication techniques, client-side voting software and secure platform mechanisms. We have provided an overview of different implementations in each category. The advantages and disadvantages with regard to security, usability, cost, and maintenance of each implementation have also been discussed. It has been shown that there are in general conflicts between these different aspects. We further recommend balancing these aspects per dimension and also between the dimensions for example, focusing on usability for one dimension and focusing on security for another one might result in a system that is neither very secure nor very user-friendly. Emphasis has also been given to the fact that not all combinations can be implemented together. Consider the following examples; a voter who selects the web browser solution is unable to also use the smart card (token) solution unless they have the required additional hardware. Use of biometrics also requires a reader. The web browser solution does not provide implementation of cryptographic protocols, taking advantage of SSL only. This is due to the fact that cryptographic implementations reside on the server side.

As there is no perfect solution for any of the dimensions one should carefully discuss the advantages and disadvantages of each of the implementations. For this discussion it is also very important to consider the environment in which the remote Internet voting system will be used and what the available techniques are in this environment. Rather than deploying smart card and biometric solutions purely for remote Internet voting, the voter and the electoral body can consider other solutions that are available, such as use of passwords. This is because the initial costs can be prohibitive. As a further example, elections carried out in a company would not best be served by the bootable CD/DVD solution, since the environment is assumed to be more secure than using a computer in an Internet café. As such, the solution deployed will greatly be determined by the money available and the environment in use as well as the appropriate combinations and existing hardware or software.

With this overview of different solutions, the discussion of their advantages and disadvantages, it should be easier for a voter or the electoral body to decide which implementation is most appropriate for their environment.

References

1. Alkassar, A., & Sadeghi, A., & Schulz, S., & Volkamer, M. (2006). Towards Trustworthy Online Voting, In Proceedings of the 1st Benelux Workshop on Information and System Security – WISSec ’06. Retrieved January 14, 2011, from https://www.cosic.esat.kuleuven.be/wissec2006/ papers/17.pdf
2. Chaum, D., & Pedersen, T. (1992). Wallet Databases with Observers. In Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, (pp 89-105).
3. Chaum, D. (2001). Sure Vote: Technical Overview. In Proceedings of the workshop on trustworthy elections (WOTE 01). Retrieved January 16, 2011, from http://vote.caltech.edu/backup/wote01/pdfs/surevote.pdf
4. Cramer, R., & Pedersen, T. (1994). Improved Privacy in Wallets with Observers (Extended Abstract). In: T. Helleseth (Ed.), Advances in Cryptology - EUROCRYPT ’93 (pp. 329–343). Springer (LNCS 765).
5. Gesellschaft für Informatik: Information für GI-Mitglieder zu Möglichen Sichereitsproblemen auf clientenseite bei wahlen mit dem onlinewahlverfahren. (2010). Retrieved January 16, 2011, from https://www.gi-ev.de/fileadmin/redaktion/Wahlen/handreichungen_gi_onlinewahlen.pdf
6. Hof, S. (2004). E-Voting and Biometric Systems. In Electronic Voting in Europe, (pp. 63-72). A. Prosser & R. Krimmer & R. Kofler (editors), Springer (LNI 47).
7. Jones, D. (2002). Trustworthy Systems on Untrusted Machines. In Workshop on the Future of Voting Technology in a Networked Environment. Retrieved January 14, 2011, from http://www.cs.uiowa.edu/~jones/voting/atlanta/
8. Juels, A., & Catalano, D., & Jakobsson, M. (2005). Coercion-Resistant Electronic Elections. In Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society – WPES ’05 (pp. 61–70). New York. ACM Press.
9. Otten, D. (2005). Mehr Demokratie durch Internetwahlen? Presentation at Nixdorf Forum in Paderborn
10. Oppliger, R. (2002*).* How to address the Secure Platform Problem for Remote Internet Voting. In Proceedings of the 5th Conference on Security in Information Systems (SIS 2002), Hochschulverlag, (pp. 153-173). Retrieved January 16, 2011, from http://pubs.esecurity.ch/sis_2002.pdf
11. Oppliger, R., & Schwenk J., & Helbach, J. (2008). Protecting Code Voting Against Vote Selling. In A. Alkassar & J. H. Siekmann (Eds.), *Sicherheit 2008*; 128, 193–204.
12. Volkamer, M., & Alkassar, A., & Sadeghi, A., & Schultz, S. (2006). Enabling the Application of Open Systems like PCs for Online Voting. Proceedings of the Frontiers in Electronic Elections – FEE ’06. Retrieved January 14, 2011, from http://fee.iavoss.org/2006/papers/fee-2006-iavoss-Enabling_the_application _of_open_systems_like-PCs_for_Online_Voting.pdf
13. Weber, J. & Hengartner, U. (2009). Usability Study of the Open Audit Voting System Helios. Retrieved January 16, 2011, from http://www.jannaweber.com/wp-ontent/uploads/2009/09/858Helios.pdf