TRUST MODELS FOR REMOTE ELECTRONIC VOTING

Melanie Volkamer [1], Rüdiger Grimm [2]

Abstract – System security requirements are supplemented by the underlying trust model containing assumptions on the operational environment and the intruder's technical capability . This is also respected by the Common Criteria and in particular its Protection Profile (PP) construct. Consequently, systems being compliant with a particular PP run securely only under the defined trust model. Therefore, a realistic description of the underlying trust model is very important. This holds particularly for critical applications like remote electronic voting. A respective Protection Profile has been certified in these days, the so called GI/BSI/DFKI PP. Some readers argue that some of the assumptions defined by this PP are not realistic in practice. Therefore, this paper discusses the consequences for the system design when changing the GI/BSI/DFKI PP trust model. In particular, the paper focuses on the assumptions that vote casting devices are trustworthy and the intruder's technical capability that cryptographic keys cannot be broken.

1. Introduction

Remote electronic voting systems are more and more used – not so much for parliamentary elections, but for elections on lower levels, for example in associations and at universities. A basis for the evaluation and certification of system security is laid by the Protection Profile [1] recently developed in Germany. Over the last two years, the Gesellschaft für Informatik (GI) has developed a Protection Profile (PP) for a basic set of security requirements for remote electronic voting systems [1] in cooperation with the German Federal Office for Information Security (BSI) and the German Research Center for Artificial Intelligence (DFKI). This GI/BSI/DFKI Protection Profile is based on the Common Criteria (CC) [2] and serves as basis for a standardised evaluation. It defines a minimum set of security objectives which every remote electronic voting system has to ensure (at least) and a maximised underlying trust model. According to the CC principles a remote electronic voting system certified against this GI/BSI/DFKI PP assures a secret, free, equal and universal election only under the condition that the system is used in an environment where the defined trust model holds. This trust model comprises a maximal set of security requirements for the environment of a system as well as a realistic estimation of the intruder's capabilities. The Common Criteria defines the trust model mainly in the security problem definition section of a Protection Profile.

A typical problem of the PP specification is the decision about the place of a security objective; either as a security objective for the system to be evaluated or as a security objective for the environment the evaluated system is used in. In general, the smaller the kernel of system security objectives, the richer becomes the trust model. The GI/BSI/DFKI

[1] Institute of IT-Security and Security Law (Passau University), 94036 Passau, Instrasse 43, volkamer@uni-passau.de

[2] University of Koblenz-Landau, 56070 Koblenz, Universitätsstraße 1, grimm@uni-koblenz.de

PP addresses only a "basic set of security requirements", consequently the defined trust model is critical, that is, the set of assumptions on the environment and the intruders' capability can be suspected to be unrealistic. Note, it is important that the trust model definition fits to the situation in which the remote electronic voting system is operated. If the corresponding trust model causes unsolvable constraints then the election authority must not implement remote electronic voting systems even if they are evaluated to be "secure". Reason: they are only secure in an environment which the election authority cannot provide.

From this point of view, this paper discusses first the importance of trust models in the context of remote electronic voting and in particular for the evaluation of related voting systems. We propose different implementation forms of two issues of the trust model, and we will discuss the consequences for the developer of a remote electronic voting system. The paper is structured as follows: In section 2, we explain how the complexity of secure systems is reduced by the implementation of a trust model about an environment. Section 3 and 4 discuss two exemplary modifications of the trust model and their consequences for the required security functions. In section 3, this is done with respect to the security objective for the environment that vote casting devices are trustworthy and in section 4, with respect to the implicitly assumption that an intruder is not able to decrypt votes. Finally, section 5 draws conclusions for the definition of the trust model.

2. Trust Models versus Secure Systems

Ideally, remote electronic voting systems are technically equipped in such a way that they enforce all security requirements. Then they would run securely in any environment, especially in insecure environments. In fact, real environments including uneducated voters, public-domain browsers, standard operating systems and unprotected open networks are naturally insecure. Theoretically, corresponding strong system security requirements can be specified. Unfortunately, developing a system in compliance with so strong system requirements is difficult, if not unfeasible. Moreover, such a system would become very complex because it would contain many more components than the pure voting protocol. It is likely that it would be too complex to be evaluated in reasonable time and under acceptable costs. This problem holds not only for electronic voting systems, but in general. Therefore, the idea is to put some constraints on the operational environment and to demand that it is not completely insecure. However, the assumptions about the environment and the intruder's capabilities must be specified explicitly. This concept is supported by the Common Criteria's "security problem definition" in combination with the definition of the "intruder's capability" by the four attacker potential values: basic, enhanced-basic, moderate, and high. Depending on the environment and the importance of the election (and thus, the expected intruders), some of the defined threats can completely, or in parts, be encountered by corresponding assumptions on the operational environment. According to the Common Criteria, the respective security objectives of the system can be shifted to security objectives of the environment. This operation can dramatically reduce the complexity of the system. On the basis of a remote voting system which is evaluated to comply with a Protection Profile of this type, the election authority in charge has to decide whether or not the defined assumptions can be realised by their environment. The same holds for the assumed intruder capability.
Therefore, the list of assumptions may vary with the type of elections. The decision whether security objectives are related to the system or to the environment is a decision about the complexity of the system security requirements versus the complexity of the trust model. Obviously, this decision has consequences for the system design and complexity. In the

following section we are going to discuss these consequences for two examples in the context of remote electronic voting.

3. Assumption of a Trustworthy Vote Casting Device

In an ideal word, a Protection Profile for remote electronic voting would contain the following security objective for the remote electronic voting system (shortcut with O.TamperClient). It needs to be ensured (in an ideal world) and is deduced from a corresponding threat (shortcut T.TamperClient):

O.TamperClient*: The client-side voting software shall ensure that its operations and data are unaffected by other applications running on the vote casting device.*

T.TamperClient: *An outside intruder runs malware on the vote casting device, which reads the vote (in order to break election secrecy), or alters the vote, or reads the authentication information to cast a vote or to bar the voter from casting a vote (in order to affect the election result).*

Now, the election authority needs to define their particular trust model and, thus, decides whether this corresponding threat is already prevented by the environment (according to the GI/BSI/DFKI PP), does not exist in their scenario at all[3], or has to be prevented by the remote electronic voting system (thus O.TamperClient holds). Depending on the decision, the following three cases can be distinguished:

Case 1: The responsible election authority can assume that voters use a trustworthy vote casting device. This could be the case, for instance, for staff and council work elections where all PCs are centrally administered and secured. On this type of vote casting devices, the voter (or any other intruder) has no possibility of installing malware on purpose and the administrator can implement adequate security mechanisms on the PCs and can ensure that all running applications do not interfere with the client-side voting software. For this case, the named threat can be reformulated to a corresponding assumption (on connectivity aspects) to the environment:

A.TamperClient: *The vote casting device is trustworthy.*

According to the Common Criteria, the security objective itself is adjusted that it does not need to be ensured by the client-side voting software but by the environment (use OE instead of O):

OE.TamperClient*: The administrator(s) of the vote casting device is responsible for its trustworthiness, that is, ensuring that other applications running on the vote casting device do not interfere with the client-side voting software, its operations, and its data as well as preventing any intruder from running malware on the vote casting device which interferes with the client-side voting software, its operation, and its data.*

Consequences: The developer of a remote electronic voting system can assume a trustworthy vote casting device and does not need to implement any security functions to protect the

[3] This can also be the case if the property for the appearance of the threat is negligible.

client-side voting software. However, such a remote electronic voting system enables secure elections only if the assumption holds for the client-side voting software, that is, if the administrator successfully secures the vote casting device.

Case 2: The responsible election authority assumes no particular attack (as described in T.TamperClient) on their election. A possible reason for such a decision can be that the effort to implement such specific malware is too much effort compared with the value of the election. Thus, the property for the appearance of the threat is negligible. However, it is well-known that various kinds of malware are available and many vote casting devices are already infected. Such malware usually tries to interfere with e-banking applications or, in general, to get user logins and corresponding passwords. To handle this remaining part of T.TamperClient, either a corresponding assumption of the environment needs to be defined (case 2A) or this part of the threat remained (case 2B):

Case 2A: The following two assumptions can be distinguished:

A.TamperClientA*: Any intruder does not try to run remote electronic voting specific malware on the vote casting device which interferes with the client-side voting software, its operation, or its data.*

A.TamperClientB: *The vote casting device is trustworthy with respect to standard vulnerabilities.*

The corresponding security objectives for the environment are:

OE.TamperClientA: *A remote electronic voting specific malware which interferes with the client-side voting software, its operation, or its data does not exist on the vote casting device.*

OE.TamperClientB: *The voter is responsible for the trustworthiness of his vote casting device with respect to standard vulnerabilities*.

Consequences: The developer of a remote electronic voting system can assume a trustworthy vote casting device and does not need to implement any security functions to protect the client-side voting software. However, such a remote electronic voting system enables secure elections only if the assumption holds for the client-side voting software, that is, remote electronic voting specific malware does not exist and the voter secures his vote casting device against standard vulnerabilities. To do so, he must have the ability to secure his vote casting device. Thus, the responsible election authority shall help voters to clean their vote casting devices from such malware. The GI, for instance, applies for their elections a simplified voters' guide [3], which contains one page of general hints and thirteen easy-to-follow one-sentence rules for voters.

Case 2B: Compared with case 2A, the security problem definition contains only the first assumption, while the second assumption is replaced by a threat:

A.TamperClient: *Any intruder does not try to run remote electronic voting specific malware on the vote casting device which interferes with the client-side voting software, its operation, or its data.*

T.TamperClient: *An outside intruder uses standard malware on the vote casting device, which reads the vote (in order to break election secrecy), or alters the vote, or reads the* *authentication information to cast a vote or to bar the voter from casting a vote (in order to affect the election result).*

From this assumption and threat, the following security objective for the environment and for the TOE can be deduced:

OE.TamperClient: *Remote electronic voting specific malware which interferes with the client-side voting software, its operation, or its data does not exist on the vote casting device.*

O.TamperClient: *The client-side voting software shall be robust against standard vulnerabilities of vote casting device.*

Consequences: The developer of a remote electronic voting system cannot assume anymore that their client-side voting software runs on a completely trustworthy vote casting device but must be aware that standard vulnerabilities still exist. Thus, the developer must demonstrate that he implements the corresponding security functionality.

Case 3: The responsible election authority considers the vote casting devices as open systems and assumes that voters are not able to protect themselves efficiently against malware. Moreover, from their point of view, it cannot be excluded that a malicious voter manipulates his vote casting device on purpose, in order to generate a proof of his choice, since a platform owner has complete control over it. Thus, the security problem definition remains as proposed in the very beginning of section 3.

Consequences: In this case, *T.TamperClient* produces a serious problem because malicious code can be distributed easily and automatically, for example, by exploiting security flaws of the vote casting device or by sending infected e-mails to voters, which could be done massively via viruses. Malicious code could also be put on the vote casting device by developers of products running on many vote casting devices (for example, Solitair). Compared to postal voting, this attack can be done automatically and in large-scale with significant impact on the election result. However, common cryptographic means do not overcome any of these two attacks, since malicious code can interact before the cryptographic operations are applied. The intruder may, for instance, eavesdrop on mouse or keyboard inputs and deduce the voter's choice. Different approaches for overcoming the weaknesses of the vote casting device have been proposed in the past, while most of them address the problem but do not satisfactorily solve it:

- The GI guidelines [3] explaining to voters how to improve the trustworthiness of their vote casting device: This approach can reduce the risks created by malware, but 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.
- Otten proposes in [4] a special voting operating system based on Knoppix. Here, voters have to boot their vote casting device from CD. This approach also does not solve the malicious voter problem, but it prevents attacks caused by malware.
- [5] and [6] proposes the application of an observer, for instance, a smart card. By doing so, they overcome 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 modified vote to the smart card.

- Helbach et. al propose in [9] (and the later improved version [10]) the code sheets to overcome the problem with malicious clients. This code sheet is sent via ordinary mail and contains for each candidate a voting TAN and a confirmation TAN[4]. The voter enters a corresponding voting TAN instead of choosing a candidate on the PC screen. To verify the correctness, he compares the received and displayed confirmation TAN with the one on the code sheet. The disadvantages of this approach concerns the user-friendliness (which decreases in particular for complex ballots implementing) and the fact that the requirement O.T.ProofGen can only be ensured if re-voting is applied.
- Another approach proposes 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 prevent the described threat. However, currently, there are still open problems with Trusted Computing and it is not easy to know how to integrate the Trusted Computing elements in a Common Criteria evaluation. For a more detailed discussion of this case and in particular the Trusted Computing based approach, see [9] and [10].

This short analysis shows that currently defining only a security objective for the TOE with respect to the client weakness would avoid the application of remote electronic voting systems because the only approach meeting the security objective is not yet implemented and ready for a large-scale application, such as in an election.

4. The Intruders' Capability to Break Encryption Keys

Depending on the concrete definition of the trust model in terms of the intruder's capabilities, the following security objective in the PP is difficult to meet:

O.ElecSecrecyNet: *The remote electronic voting system shall not provide any information in transmitted protocol messages, which allow one to construct the link between a particular voter and his vote.*

T.ElecSecrecyNet: *An outside intruder reads on the network in order to break the election secrecy.*

With respect to this security objective, the responsible election authority needs to decide ...

Case 1 ... whether it is acceptable that the intruder is able to break the election secrecy after a particular point in time (for instance, after the next election). The consequence is that the named threat needs to be extended in the following way:

T.ElecSecrecyNet: *An outside intruder reads on the network in order to break the election secrecy before the next election.*

A similar extension needs to be added to the security objective:

O.ElecSecrecyNet*: The remote electronic voting system shall not provide any information in transmitted protocol messages, which allow one to construct the link between a particular voter and his vote before the next election.*

[4] To overcome vote selling the authors introduced in [10] an additional TAN – the so called finalisation TAN.

Case 2 ... whether it is acceptable that the intruder is able to break the election secrecy (either before or after the next election) as long as he cannot prove the link. The consequence is that the named threat needs to be changed in the following way:

T.ElecSecrecyNet: *An outside intruder reads on the network in order to break the election secrecy and is able to prove the link between the voter and his vote.*

A similar modification needs to be added to the security objective:

O.ElecSecrecyNet: *The remote electronic voting system shall not provide any information in transmitted protocol messages, which allow one to construct the proof for the link between a particular voter and his vote.*

Case 3 ... whether only those remote electronic voting systems are acceptable, which ensure that the voter can never be linked to his vote by an outside intruder reading on the network (this would be in compliance with the temporal unlimited election secrecy demanded in [9]). Thus, the security problem definition remains as proposed above on page 8 at the beginning of this section.

Feasibility consideration. In general, it is not possible to prevent the reading on the network even sniffing and data decryption are punishable according to § 202aStGB in Germany and corresponding laws in other countries. The intruder works in the following way. He sniffs all voting protocol messages transmitted to the voting server, stores these data in a database and analyses them later. These messages are encrypted with state-of-the-art encryption algorithms which are classified as secure. The problem with respect to the security objective O.ElecSecrecyNet is that the chosen algorithms might be classified as secure for the present and possibly also for the near future, but no statements for the long future can be made. Perhaps, someone will find a fast algorithm to decrypt messages without the knowledge of the secret key, allowing to break the applied cryptographic algorithm. In any case, by using adequate computational power, single messages can be decrypted or single secret keys can be calculated (brute force trials). Depending on the intruder's computational power, the intruder will be in a position to decrypt all or some encrypted ballot messages at some time in the future. Thus, it cannot be prevented that the intruder will be able to decrypt these messages at some time in the future. However, to ensure O.ElecSEcrecyNet [case 1] the application of state-of-the-art encryption algorithms would work. The question for the other two cases is, whether the intruder is able to link the decrypted vote message to the corresponding voter ID or whether he only gets decrypted vote messages but cannot link these to voters. The analysis of different types of remote electronic voting systems in [11] shows that temporal unlimited election secrecy like demanded in O.ElecSecrecyNet [case 3] cannot be ensured by any of the analysed remote electronic voting systems because of bindings to voter's IP-address, or even more, because of the binding to the voter's digital signature. Exceptions are those remote electronic voting systems that implement re-voting like in Estonia or those that implement a two-phase voting protocol (see, for instance, [9]). [11] also points out that, in general, there is no possibility to prove the knowledge to a third party, which enables the application of O.ElecSecrecyNet [case 2].

5. Conclusion

This contribution underlines the importance of the trust model definition for any kind of system evaluations and in particular for evaluations of remote electronic voting systems according to the Common Criteria. Thereby, the trust model definition is important to decide whether a particular system is compliant to the defined security requirements. Such a statement is not possible for the general case, but only under the conditions of the defined trust model. Vice versa, the evaluated system does only ensure the security requirements if the trust model holds for the environment the system is used in. Therefore, it is essential to base the evaluation on a trust model that really holds in the environment. However, changing or strengthening the trust model has consequences for the system design. This contribution shows that for some classes of trust models for remote electronic voting systems, there exists (currently) no possibility to develop a corresponding remote electronic voting system that is compliant to the security requirements under these trust models. In these cases, remote electronic voting systems should not be used.

Note that it is the task of the responsible election authority to decide if the trust model realistically fits to the application environment. In particular, the election authority must respond to the two questions discussed in this contribution: how long must the election secrecy be ensured and how trustworthy is the vote casting device? As shown in the previous section, depending on their decision, the system has to provide more or less security functions. In one of the discussed cases a corresponding implementation is even not possible.

References

[1] M. Volkamer and R. Vogt. Basissatz von Sicherheitsanforderungen an Online-Wahlprodukte, **Fehler! Verweisquelle konnte nicht gefunden werden.**, http://www.bsi.de/cc/pplist/pplist.htm, 2008.

[2] Common Criteria for Information Technology Security Evaluation, Version 3.1, http://www.commoncriteriaportal.org/thecc.html, 2006.

[3] Information für GI-Mitglieder zu möglichen Sicherheitsproblemen auf Clientseite bei Vorstands- und Präsidiumswahlen mit dem Online-Wahlverfahren. Technical report, Gesellschaft für Informatik and F-Secure Deutschland, 2005.

[4] D. Otten. Mehr Demokratie durch Internetwahlen? Vortrag gehalten im Nixdorf Forum in Paderborn, 2005.

[5] J. Schweisgut. Coercion-Resistant Electronic Elections with Observer. In R. Krimmer, editor, Electronic Voting 2006, volume 86 of LNI, pages 171–177. GI, 2006.

[6] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In WPES ’05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society, pages 61–70, New York, NY, USA, 2005. ACM Press.

[7] J. Helbach and J. Schwenk. Secure Internet Voting with Code Sheets. In A. Alkassar and M. Volkamer, editors, Vote ID 2007, VOTE-ID 2007, volume 4896 of LNCS, Springer-Verlag, pages 166–177, 2007.

[8] R. Oppliger, J. Schwenk, and J. Helbach. Protecting Code Voting Against Vote Selling. In A. Alkassar and J. Siekmann, editors, Sicherheit, volume 128 of LNI, pages 193–204, Bonn, GI, 2008.

[9] A. Alkassar, A.-R. Sadeghi, S. Schultz, and M. Volkamer. Towards Trustworthy Online Voting. Proceedings of the 1st Benelux Workshop on Information and System Security (WISSec 2006), 2006.

[10] M. Volkamer, A. Alkassar, A.-R. Sadeghi, and S. Schultz. Enabling the Application of Open Systems like PCs for Online Voting. Frontiers in Electronic Elections (FEE 2006), 2006.

[11] R. Kofler, R. Krimmer, and A. Prosser. Electronic Voting: Algorithmic and Implementation Issues. In HICSS ’03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS’03) - Track 5, page 142.1, Washington, DC, USA, 2003. IEEE Computer Society.

[12] M. Ullmann, F. Koob, and H. Kelter. Anonyme Online-Wahlen - Lösungsansätze für die Realisierung von Online-Wahlen. DUD * Datenschutz und Datensicherheit 22, 2001. V2.6.

[13] M. Volkamer and R. Krimmer. Secrecy forever? Analysis of Anonymity in Internetbased Voting Protocols. In ARES ’06: Proceedings of the First International Conference on Availability, Reliability and Security (ARES’06), pages 340–347,Washington, DC, USA, 2006. IEEE Computer Society.