FIDIS

Future of Identity in the Information Society

Title:	"D16.4: Study on the relevance of Trusted Infrastructures for E-Voting"
Author:	WP16
Editors:	Rani Husseiki (Sirrix AG)
Reviewers:	Melanie Volkamer (CASED), Marcel Winandy (Ruhr University of Bochum)
Identifier:	D16.4
Type:	[Deliverable]
Version:	1.0
Date:	Friday, 14 August 2009
Status:	[Final]
Class:	[Public]
File:	fidis_wp16_d16.4_v1.0.doc

Summary

In this deliverable, the implications of Trusted Infrastructures on e-voting are assessed. This is done by defining a trust model for three different e-voting schemes (machine voting, internet voting, and sms voting), and assessing how Trusted Infrastructure concepts (such as Trusted Computing functionalities) are able to address the underlying assumptions which are necessary for a secure, reliable and trustworthy e-voting process. In particular, we emphasize conflicting requirements such as anonymity vs. authentication, and receipt-freeness vs. vote verifiability. These identity-related requirements entail special technical approaches that Trusted Infrastructures can provide. We also shed the light on the social and legal implications of deploying a Trusted Infrastructure to support an e-voting system.

The FIDIS NoE receives research funding from the Community's Sixth Framework Program

Copyright Notice:

PLEASE NOTE: This document may change without notice – Updated versions of this document can be found at the FIDIS NoE website at www.fidis.net.

Members of the FIDIS consortium

1.	***Goethe University Frankfurt***	Germany
2.	***Joint Research Centre (JRC)***	Spain
3.	***Vrije Universiteit Brussel***	Belgium
4.	***Unabhängiges Landeszentrum für Datenschutz (ICPP)***	Germany
5.	***Institut Europeen D'Administration Des Affaires (INSEAD)***	France
6.	***University of Reading***	United Kingdom
7.	***Katholieke Universiteit Leuven***	Belgium
8.	***Tilburg University***[1]	Netherlands
9.	***Karlstads University***	Sweden
10.	***Technische Universität Berlin***	Germany
11.	***Technische Universität Dresden***	Germany
12.	***Albert-Ludwig-University Freiburg***	Germany
13.	***Masarykova universita v Brne (MU)***	Czech Republic
14.	***VaF Bratislava***	Slovakia
15.	***London School of Economics and Political Science (LSE)***	United Kingdom
16.	***Budapest University of Technology and Economics (ISTRI)***	Hungary
17.	***IBM Research GmbH***	Switzerland
18.	***Centre Technique de la Gendarmerie Nationale (CTGN)***	France
19.	***Netherlands Forensic Institute (NFI)***[2]	Netherlands
20.	***Virtual Identity and Privacy Research Center (VIP)***[3]	Switzerland
21.	***Europäisches Microsoft Innovations Center GmbH (EMIC)***	Germany
22.	***Institute of Communication and Computer Systems (ICCS)***	Greece
23.	***AXSionics AG***	Switzerland
24.	***SIRRIX AG Security Technologies***	Germany

[1] Legal name: Stichting Katholieke Universiteit Brabant
[2] Legal name: Ministerie Van Justitie
[3] Legal name: Berner Fachhochschule

Versions

Version	*Date*	*Description (Editor)*
1.0	20.06.2009	• Initial release (Rani Husseiki)
		•

Foreword

FIDIS partners from various disciplines have contributed as authors to this document. The following list names the main contributors for the chapters of this document:

Chapter	*Contributor(s)*
1 (Executive Summary)	Rani Husseiki (Sirrix AG)
2 (Introduction)	Rani Husseiki (Sirrix AG)
3 (E-voting-specific Requirements)	Eric Dubuis (VIP)
4 (An Overview of Trusted Infrastructures)	Rani Husseiki (Sirrix AG)
5 (Trusted Infrastructures for e-voting)	Eric Dubuis (VIP), Rani Husseiki (Sirrix AG) and Yianna Danidou (University of Edinburgh)
6 (Analysis)	Rani Husseiki (Sirrix AG), Stefan Berthold (TUD) and Yianna Danidou (University of Edinburgh)
7 (Summary)	Rani Husseiki (Sirrix AG)

Table of Contents

1 Executive Summary

In this deliverable, we laid down the general requirements for E-voting, and emphasized the requirements that are conflicting among each other in some aspects. Specifically, the anonymity of the vote (a fundamental requirement entailing non-linkability between the identity of the voter and his vote) and the eligibility for voting (another fundamental requirement entailing authentication based on identification of the voter) are inherently in conflict. Another case is that of the receipt-freeness requirement (necessary to avoid vote coercion and vote trade) and the individual vote verifiability requirement (necessary to assure the voter that his selection has been tallied correctly).

Trusted Infrastructures concepts, which are explained in chapter 4, include a set of mechanisms, components and protocols that are combined and adequately used to provide a certain level of trust in IT infrastructures. Trusted Computing, a key concept underlying Trusted Infrastructures, is based on the concept of integrity measurements, and relies on a TPM, a tamper-resistant security chip. The TPM integrates the core root of trust in a system, and allows encrypted software fingerprints stored inside it to be communicated via remote attestation protocols to a remote verification party. Based on this concept, trusted channels are designed, which basically are secured channels established between trustworthy endpoints. The features of Trusted Infrastructures motivated the work on studying their relevance to e-voting schemes especially with regard to the problem of insecure voter device in internet-voting schemes.

Three different kinds of e-voting schemes were analysed in chapter 5, namely machine-voting, internet-voting, and sms-voting. For each of the three schemes, a description, a trust model and the Trusted Infrastructures supporting concepts are explained with regard to how they can address the trust assumptions of the scheme. This analysis showed that Trusted Infrastructures can make a considerable impact on e-voting systems in various ways to fulfil the trust assumptions underlying a reliable, trustworthy and secure e-voting scheme.

The analysis in chapter 6 showed that some of these Trusted Infrastructure concepts (e.g. small and verifiable TCB, TPM, integrity measurements) are relevant and apply similarly to the three discussed kinds of e-voting. Other concepts, e.g. combination of virtualization and TC techniques such as remote attestation and trusted channels can be used selectively, in different manners and for different purposes according to each of the three schemes. In any case, the deployment of a Trusted Infrastructure for e-voting assumes that voting devices, whether public or proprietary, must include e-voting software with specific characteristics, such as verifiability, non-complexity and modularity. This is necessary to verify the compliance of the software against trustworthy specifications ahead of elections. Moreover, the involved devices must include specific software (e.g. security kernel with virtualization capability) and hardware (TPM chip).

Nevertheless, when deployed for an e-voting system, Trusted Infrastructures should not be expected to add more security to the voting process as compared to traditional voting schemes. They must be deployed however in order to address the additional trust requirements introduced through the use of technology.

We also summarized the requirements for a socially acceptable e-voting system with Trusted Infrastructure support. They entail rigorous specifications and standards for the e-voting software to be public available for inspection, preventing fraud, attacks, and privacy violation, and certified by independent bodies. Trusted Infrastructures help verifying the compliance to these certifications by the voters and voting server administrators, and can therefore give confidence to each of the parties that the level of trust in the technical infrastructure is high enough despite the use of technology.

We also discussed a couple of alternatives to machine and internet e-voting systems that present a hybrid approach combining traditional (paper-based) approaches with machines, namely the Prêt-à-Voter scheme and the Scantegrity scheme.

2 Introduction

As E-Voting becomes a stressing necessity, with a growing inclination from people to vote using their proprietary devices, sophisticated techniques are needed in order to establish trust in an infrastructure supporting the e-voting process. Identification of voters eligible to vote, preserving anonymity of the vote, avoiding receipts of vote confirmation messages that can be used in vote coercion and trade are few of the requirements that necessitate a new approach considering the replacement of administrative personnel and election officials by technology.

Trusted Infrastructures have emerged as a fundamentally new approach for achieving trust in a distributed system. It relies on the notion of distributed trust model and trust propagation, and integrates state-of-the-art security technologies, such as Trusted Computing and Public Key Infrastructures, in order to achieve assurance in the reliability and security of remote components of the system.

In this deliverable, we discuss the possible deployment of Trusted Infrastructure technologies to enhance, its implications on the general trust models of several e-voting schemes. The motivation behind the work is that Trusted Computing, which is based on a specific tamper-resistant hardware module (TPM), can be deployed as a means for integrity attestation that would allow the voter's device and the E-voting Server to exchange attestation certificates. Those certificates would prove the integrity, eligibility and standard configuration of the software running on the server or voter's device. This way, the user is assured that the server is not bogus and that his vote is not revealed. On the other hand, the server would be assured that the voter's device is not compromised (e.g. by malware) or its software manipulated.

In particular, we address the machine-voting, internet-voting, and sms-voting schemes. For each of these schemes, we lay down the trust assumptions for ensuring a correct, fair and secure voting process. Then we try to investigate how Trusted Infrastructure concepts can be implemented in a way to address these assumptions, and where these concepts can fall short from fulfilling certain assumptions which can only rely on other factors, such as trustworthiness of election officials.

In other words, the approach we follow is:

1) Studying the requirements for a generic E-Voting scheme, such as identification, verifiability and anonymity, and presenting an overview of the available schemes used, along with their advantages and disadvantages.

2) Investigating the technical problems that can be solved by Trusted Infrastructures deployment based on Trusted Computing technology, emphasizing the Trusted Computing functionalities that can help solve those problems.

3) Studying the possible implications of the use of Trusted Computing, how it can achieve strong identification of people, if it would enhance the privacy or anonymity of the vote, how would identities credentials be supported and the profiling resulting from a multiple-vote scheme.

In particular, we emphasize the relevance of Trusted Infrastructures with respect to specific conflicting requirements in general e-voting schemes. For example, one problem in focus is that during the E-voting process, the voter has to reveal his identity by supplying necessary credentials through his device in order to prove his eligibility to contribute to the vote. At a second stage, the choice of the voter is supplied to the device, but should not be associated to the voter identity since many of the E-Voting schemes require complete anonymity. Therefore, we investigate how the trade-off between reliable identification and voting anonymity can be handled with the deployment of Trusted Computing functionalities for the E-Voting scheme, aiming at:

- Enhancing the identification of citizens through their proprietary devices as to make the voting body confident about the trustworthiness of the device used for voting.
- Preserving the privacy of the vote in a way to make the voter confident that an association between his identity credentials and his vote is impossible.

We also shed some light on the implications of deploying Trusted Infrastructures for the sake of e-voting on the voting citizens.

At the end, a couple of alternative solutions to machine and internet voting are presented. Those integrate a hybrid approach by combining traditional voting schemes with machines.

3 E-voting-specific Requirement

It is widely recognized that the requirements for e-voting systems must be derived from a number of obligations and commitments that are laid down in documents such as the Universal Declaration of Human rights, the International Covenant on Civil and Political Rights, the national constitutions, to name a few. These documents form the basis on which the more specific e-voting requirements can be based on.

The e-voting requirements can be partitioned into the following group of requirements [1, 2]

- requirements on democratic aspects
- requirements on privacy
- requirements on verifiability
- requirements on fairness

Depending on the literature source, different definitions can be given to each of the requirements. In the following sections, we list the definitions that are the most widely acknowledged.

3.1 Democracy

The requirements on democratic aspects can be further separated into more detailed requirements.

3.1.1 Exclusiveness

(R1) Exclusiveness: An e-voting system is democratic if only eligible citizens are exclusively authorized to vote.

The e-voting system must have a means to authenticate a citizen. Only after a successful authentication does the e-voting system give the authenticated citizen the right to cast a vote.

3.1.2 Eligibility

(R2) Eligibility: everyone who is eligible to vote should have the possibility to do so.

The e-voting system should allow all citizens who have the right to vote to have access to it without discrimination. This means that no eligible voter should have an advantage in the possibility or ease of accessing the system over another eligible voter.

3.1.3 Uniqueness

(R3) Uniqueness: An e-voting system is democratic if and only if eligible voters can vote only once.

The e-voting system must have a means to avoid that an eligible voter can vote more than once. In other words, if the e-voting system has permitted an eligible voter to cast his/her vote for the first time, it must prevent the casting of the second vote.

3.1.4 Final Vote-counting consistency

This general requirement consists of three parts.

(R4) Integrity: An e-voting system is democratic if and only if a vote being cast cannot be altered and no unauthorized vote can be added to the electronic ballot or valid vote removed during and after the voting process.

An e-voting system must ensure that a vote being cast resists to undergo any modification neither while in transit to the e-voting system, nor while the vote is stored wherein, and nor while it is counted. It should also be impossible to add a vote to the ballot violating the first three requirements, e.g., by manipulating the system.

(R5) Completeness: An e-voting system is democratic if and only if a validated vote being cast and correctly stored cannot be eliminated from the final tally.

An e-voting system must ensure that valid votes that have been stored in the electronic ballot box are counted . If a vote could be removed undiscovered then the final tally, even if counted correctly, would not express the will of the eligible voters.

(R6) Soundness: An e-voting system is democratic if and only if an invalid vote being cast is not counted in the final tally.

An e-voting system must ensure that invalid votes are not counted. Any invalid vote in the electronic ballot box must be eliminated prior the counting takes place.

3.2 Privacy

The requirements in this section ensure the privacy of the voter.

3.2.1 Non-linkability between vote and voter (anonymity)

(R7) Anonymity: An e-voting system protects privacy if the content of a vote being cast can not be linked to its voter, neither by voting authorities nor by anyone else.

An e-voting system must ensure that there is no way to associate a vote to the citizen having cast the vote. This requirement includes the secrecy requirement that states that it must be impossible for others to know how a voter has voted, neither during the voting process nor at a later stage.

3.2.2 Non-provability of the vote (receipt freeness)

(R8) Receipt-freeness: An e-voting system protects privacy if no voter can prove that he or she voted in a particular way.

An e-voting system must not return any evidence to the voter which expresses the particular way the voter has cast his/her vote (or the system can return a receipt for any other possible vote, regardless of the choice of the voter). If it did, then a voter could sell his/her vote to any third party and prove afterwards how he/she voted.

A second problem would be vote coercion: A voter could be coerced by any third party to vote in a particular way by asking the voter to prove how he/she voted.

3.3 Verifiability

These requirements allow the voter to accept the outcome of a vote.

3.3.1 Individual vote verifiability

(R9) Individual Verifiability: A system is individually verifiable if a voter can independently verify that his/her own vote has been counted correctly in the final tally.

An e-voting system must allow the voter to convince himself/herself that his/her vote is tallied correctly.

3.3.2 Final vote-counting verifiability

(R10) Universal Verifiability: An e-voting system is universally verifiable, if a voter or others can independently verify that all casted votes have been counted correctly in the final tally (contains all votes).

An e-voting system must allow the voter to convince himself/herself that all eligible votes are tallied correctly.

3.4 Fairness (revealing non-final results)

(R11) Fairness: An e-voting system is fair if no intermediate results can be obtained before the voting period ends.

This requirement ensures that the outcome of a vote is not influenced by the votes being cast during the vote casting process.

3.5 Comments on the Requirements

Comments on the above requirements are in place. Some requirements are complementary and don't interact with each other such as the requirements on eligibility (R2) and on uniqueness (R3). Other requirements seem to be in conflict:

- Requirement (R1) states that a voter must be authorized to cast a vote. This means that the e-voting system "knows" something about the voter. This might be the voter's name, a voter's identification number, or some other means to authenticate the voter.

On the other hand, requirement (R7) states that it must be impossible to link a voter's ballot to the voter. Requirements (R1) and (R7) are inherently in conflict!

- Requirement (R8) states that an e-voting system must be receipt-free. That is, the voter doesn't receive any information which allows him/her to prove the way he/she voted. On the other hand, requirement (R9) states that a voter should be able to verify that his/her vote has been counted correctly.

The two comments given above show that the requirements seem to be in conflict, in the sense that both can not be fulfilled to a high degree, but rather a trade-off might be needed. Some authors speak of the “security dilemma” in e-voting systems. It is interesting to note that Ronald L. Rivest showed that when using three ballots, the second dilemma, the conflicting of requirements (R8) and (R9), can be avoided [3].

It must also be noted that the security requirements for e-voting systems are fundamentally different and more difficult to satisfy than those of e-commerce. In e-commerce, performed on-line financial transactions can always be audited and checked off-line, and to correct them in case of errors. This is not the case with e-voting system, since according to requirement (R8) a voter cannot proof the way he/she has voted. To put it differently, a successful attack to an election is one which, by definition, cannot be detected.

4 An Overview of Trusted Infrastructures

Trusted Infrastructures refer to IT infrastructures that able to provide a certain level of trust in the reliability and security of its components according to the expectations of the users interacting with the infrastructures. The concepts underlying Trusted Infrastructures rely on the idea of trust models.

The terminology used in trust models is similar to that of security models. There are objects and subjects. Subjects can be individuals or organizations. Trust is "propagated" through a system. Propagation is done by means of technology, for example authentication technology like cryptographic authentication protocols. This introduces cryptographic keys into trust models. Keys are used as part of certificates. Trust metrics are methods that deliver a measure of trust in certain system properties. Numerous methods and approaches exist, ranging from simple scoring up to probabilistic calculations. Trust metrics are mentioned for completeness, but will not be presented in depth in this text.

Trust is on a gradient, i.e. there are no one-size-fits-all solutions:

- Trust requirements must be matched to the specific kinds of threats or vulnerabilities facing an organization and to the degree of risk that the threats will occur.
- There must be a starting point in establishing credentials for identity. A common electronic form of such a credential is called "certificate". A certificate is created using cryptography. It contains a public key that matches the certificate's owner's secret key. A certificate is validated by a certificate authority with electronic signatures.
- Trust does not happen spontaneously. It requires a methodical process of credential establishment and consistent validation.

Another core concept of Trusted Infrastructures is the idea of "Trust propagation" which is an approach where trust is not implicit in a trust model, but can explicitly be assured to another entity. A very simple form of trust propagation is the use of passwords to log on to a computer. The possession of the password shows to the system that the user is trusted by the system administrator. In a Trusted Infrastructure, the trust in the behavior of security-critical components is in focus, and is usually propagated by means of secure channels, i.e. channels that preserve the security and confidentiality of data.

The transition from a local network to interconnected, distributed systems, e.g. on the Internet, requires thorough refinement of the trust models used. While all trusted components in a local network are subject to physical security as well as local administrative control, the remote systems in the distributed scenario can not be trusted by definition. Some form of reconnaissance mechanism, such as trust certification, is needed to distinguish trusted from untrusted systems. In this chapter, we explain the core concepts underlying Trusted Infrastructures, and emphasize the need and means for propagating trust in a Trusted Infrastructure.

4.1 Trusted Computing

Trusted Computing (TC) is a platform and security protocol specification made by the Trusted Computing Group. Its purpose is to provide a hardware-secured component and a set of secure algorithms to computer systems that can establish a security anchor for secure system startup and trust enforcement. A Trusted Computing Base (TCB) which is usually defined in the context of TC, is a subsystem of the overall computer system that enforces the above principles in an effective way. Its effectiveness shall be subject to evaluation by independent experts. In the following, we explain the basic functions of TC.

4.1.1 Integrity Measurements

Integrity measurement is one of the basic and probably most fundamental mechanisms of Trusted Computing. Integrity measurement means to calculate and store the state of a computer or device under secured conditions.

The process of measurement while booting is called trusted booting (sometimes also referred to as authenticated booting; however, this term has been used recently for a subset of trusted booting where the measurement is reported to a remote third party). It is a method that securely logs which software is booted on a computing device but does not influence the boot sequence (e.g. in deciding which software to execute and which not). After finishing the boot sequence, these logs can be used for checking the state of the system. An important remark is that trusted booting by no means ensures that the computing device is in a "secure" state.

With the help of the log entries it is also possible to report the state of the computing device to remote entities. By analyzing the logs remote entities can decide about the trustworthiness of a system. This approach is particularly suitable for secure open platforms, which can be modified in many ways.

4.1.2 Authenticated Booting (or Trusted Booting)

The technology of trusted booting has been introduced by Arbaugh et. al. in [ArFaSm97] (there, they call it authenticated boot vs. secure boot) long before the TCG specification and is already used in other designs. Currently available designs of trusted booting require new hardware or support untrusted applications insufficiently.

There exists another type of booting a system in a trusted way called secure booting. Secure booting checks the source code before executing it according to a set of security policies and thereby avoids that malicious or unauthorized source code is running on the computing device. The security policy comprises identifiers of authorized modules. Hash values can be used as identifiers. If an identifier or hash value is not found in the list of the security policy the process of booting is discontinued. Usually this technology is used to ensure locally that the platform is in a secure state, but not to prove this to a remote party. The technique of secure booting can be used to construct secure closed platforms, which have a limited number of executable software. Designs for secure booting have been known for over 15 years.

Both mentioned methods can be combined. After measurement of integrity the results are sent to a remote verifier that checks the results. If the computing platform is in an invalid state the remote verifier may initiate a remediation. So the platform has to be updated. After that it starts the integrity measurement again until it is in a valid state.

When a boot sequence can be validated (remote or locally) it ensures that components of the platform are not emulated, so that a specific hardware with a specific OS with a specific GUI and a specific application is indeed running in the identified device.

4.1.3 Binding, Sealing and Attestation

The hardware integrated root of trust can provide methods to bind data, licenses and user authentication to a specific platform which consists of specific hardware and software executed on that hardware. This can be realized by cryptographic operations which are executed and stored in a protected hardware environment. Within the TCG specification, a unique key, that is certified by the manufacturer and stored protected in the microcontroller, is essential for verifying the platform as trusted and can also be used to authenticate a special user accurately. This key is the initial point for certificates and further keys.

Sealing allows tying data to a specific computing device and thus, to restrict the access to sensible data which is stored on its hard disk to systems with a dedicated hardware and software configuration.

A major security problem of computing systems is storing cryptographic keys securely. Keys or passwords that protect private documents are often retained locally on a hard drive of a PC, side by side with the encrypted documents themselves. Everyone who gains access to the computing system can also access cryptographic keys and passwords stored. But keys should be kept secure so that only legitimate users can access them.

The technique of sealed storage is based on a key that is partially generated by the identity of the software requiring the key. Furthermore the identity of the computing device that executes the software presents the second part of the key. So, these keys need not be stored on the hard drive but can be created whenever they are required.

If a program different from the program which initiated the encryption (or sealing of sensible information) would try to decrypt or unseal this data, this fails because the generated key is not equal to the original one. That follows from the different identifiers of the software that seal and unseal the data and consequentially the generated keys are different as well. A similar use case is that encrypted data is transferred to another computing device that tries to decrypt the data. This will also be unsuccessful. So, for example emails that you can read on your computer are unreadable on other computer systems.

With the help of sealed storage one cannot prevent that confidential data is copied to another system but you can prevent others from reading it on this system. By using attestation it is possible to check the hardware and software states of a remote platform. Therefore the results of integrity measurements, which characterize the software and hardware environment of a computing system, are signed with a key. The signed outcomes can be verified by a remote party and needs no physical presence. Together with the signed outcomes certificates are sent that accredit the used key as trusted. A remote attestation can be conducted directly or through a trusted third party that verifies the remote platform as trusted.

A trusted third party (TTP) checks keys and certificates of a computing device. If they are valid, the trusted third party issues a certificate that attests a computing device as trusted.

The relation between a certain computing device and a certificate provided by a TTP should be hidden for anonymous usage.

Direct anonymous attestation (DDA) without using a third party verifier is a further attestation technique. It can be proven that a valid certificate exists without disclosing it. So certificates could be generated anonymously. Group signature schemes allow that every member of a group can sign messages on behalf of the whole group. This supports the anonymity of the group members and provides a valid signature.

4.2 Distributed Trust Requirements

As mentioned above, trust in distributed systems requires different approaches as local trust in a single system. Trusted Infrastructures, which are distributed systems, are made up of Several Machines and have some properties such as Remote Users, Untrusted communication networks, and Mobility. Trusted Infrastructures can be seen as extending the access control problem to a network of collaborating systems. Therefore, corresponding trust models are based on vital building blocks such as Secure Network Connections, Remote Credentials and Remote Integrity. In the following, we present two vital techniques for achieving trust in the distributed Trusted Infrastructures.

4.2.1 Key management

Key management is the core measure for trust in distributed systems. Cryptographic keys and algorithms fulfill many tasks in trust establishment. They are used for the secure connections, for user authentication, integrity checking and remote systems identification. Key management is based on a "Public Key Infrastructure" (PKI) which is a setting where public parts of keys can be openly distributed, but can be used to perform encryption or verification operations involving a secret key. A PKI has at least one certificate authority (CA) which is responsible for the issuance of certificates. The certificates contain certified public keys, and possibly additional data. Certificates can be used in cryptographic protocols to establish secure communication channels, encrypt and sign data or to authenticate entities. Certificates can be used in many ways to add trust to a distributed computer system, for example to establish secure communication channels, authenticate remote entities (e.g. user identities using tokens), and verify integrity of data or program code.

However, securing an infrastructure that uses secure and trusted hardware, runs trusted (certified) operating systems and applications that can all be authenticated based on certificates is an enormously complicated task. CAs have to be synchronized, and every installation and patch that might happen on any component in the distributed system might need new certificates. There certificates in turn then have to be distributed or made available to the other entities.

4.2.2 Remote Attestation

Remote attestation is another important technology for trust in distributed systems. Remote attestation (RA) provides the capability to know with certainty the hardware and software configuration of a remote host. It is a cryptography-based approach that was introduced with

Trusted Computing. Its base is a hardware support for the verification of loaded software and software that allows for remote inquiries about the state of the system. Using certificates and signatures, a remote system can not lie about its software configuration. While this capability has received much criticism of concerned anti-DRM activists, its benefits are hardly expressed with equal passion. A system of remote attestation can verify that a remote computer indeed has a particular version of a web shop system installed that is known for being privacy-conformant.

4.2.3 Trusted Channels

Trusted Channels [Gasmi, et al.] are a variant of secure channels which take into account the establishment of trust between the endpoint of the channel. This trust is established upon exchanging integrity measurements of the security-critical components of the corresponding systems, e.g. ahead of exchanging keys for securing the channel. Integrity measurements can be performed according to the TC functionality, and remote attestation techniques are typically used to exchange these measurements and verify them by the endpoints. Therefore, Trusted Channels are a mandatory requirement for establishing communication channels between distributed components of a Trusted Infrastructure.

5 Trusted Infrastructure for e-Voting

Some security requirements can be addressed with existing security technologies. For example remote e-voting systems, the eligibility (R2) can be addressed with secret codes transmitted off-line on personalized voting cards to the voter. Or, the eligibility of a voter can be verified by making use of private keys and public key certificates. Similarly, one can use SSL/TLS technologies for remote e-voting systems for ensuring the integrity (R4), of the ballot. However, it is important to note that the integrity can only be ensured while the ballot is in transit between the voter and the electronic ballot box. A ballot's integrity, however, cannot be ensured with the use of SSL/TSL technologies while being in the memory of the voter's computer or the e-voting server. In fact, additional mechanisms are needed to protect the anonymity and integrity of a ballot.

Therefore, it is clear that while many security requirements for e-voting can be handled by legacy cryptographic protocols and mechanisms, other requirements need special techniques that are not widely deployed nowadays. In particular, establishing trust in the voting machines (whether machines in polling stations, proprietary voters’ devices, or e-voting servers), entails a set of trust requirements. For example, if a voter wants to use his proprietary device to connect to an e-voting server, how can he verify the authenticity of the server? On what basis would he verify the trustworthiness of the server with regard to the democracy and privacy requirements stated in chapter 3? If public voting machines on polling stations are used, how can the voter verify the trustworthiness of the machine in front of him? On the other hand, how can the server verify the trustworthiness of the proprietary voter’s device which might be infected with a virus?

In this chapter, we address these issues which are particularly important for e-voting, by investigating the relevance of Trusted Infrastructure concepts (chapter 4) towards fulfilling the underlying trust requirements. Hence, we focus on three types of e-voting schemes: machine voting, internet voting and sms voting. For each of these schemes, we give a short description and lay down the trust assumptions that need to be guaranteed for a secure, democratic, privacy preserving, and verifiable e-voting system. Then, we investigate which Trusted Infrastructure concepts and techniques can help fulfil which of those trust assumptions.

5.1 Direct Recording Machines

5.1.1 Description

So-called *Direct Recording Machines* (DRE) are computers installed at polling stations. This type of equipment allows the voter, after having been identified to be eligible to vote, first to enter his/her voter access card received from election officials, and then to fill out a ballot displayed at the computer's screen, typically via the computer's touch screen or buttons. Once filled-out, the computer processes the voter's ballot and records it in memory. After the election the computer tabulates the voting data and prints a copy.

Voting machines used for *kiosk voting* transmit individual votes to a central voting server thus avoiding the need to record votes locally. These machines are typically installed in official buildings where the personal is able to register and check the eligibility of a person to vote, thus guaranteeing the traditional safeguards such as checking the eligibility of the voter to vote, the free expression and the privacy, of the vote. Common to DRE and kiosk voting machines is the fact that these machines are deployed and operated by election officials.

The lack of trust in these machines by voters, among other things, has led countries like Netherlands, Ireland and Germany to avoid using them anymore.

5.1.2 Trust Assumptions

For *DRE computers* installed at polling stations, trust is based on the following assumptions:

1. There exists a rigorous specification for the behaviour and security requirements, thus defining implicitly or explicitly a behavioural and security model. The requirements are publicly available.
2. The specification has been reviewed by governmental bodies or, alternatively, by recognized standardization bodies.
3. The DRE computers are built according to these behaviour and security requirements. Hardware and software components together implement the features specified in the requirements. The implementation (hardware and software) is thoroughly documented.
4. Afterward modification of the hardware components and software is defeated by introducing a sort of seal.
5. In the case where the machine is in the form of a standalone computer, the computer's box mechanically locked.
6. One or more independent certification centers analyze and test every DRE computer. This is done by explicitly checking and analyzing the implementation documents, the hardware, and the software. DRE computers are tested. The findings are documented. If the analysis and the test results confirm that the DRE computer fulfils the behaviour and security requirements (which are certified as trustworthy in (2)) then this fact is stated in the form of a certificate associated with this particular DRE computer.
7. Electoral offices buy and deploy only certificated DRE computers. They also receive the keys and have a key management in operation. (This resembles the management of the keys being used for the classical ballot boxes.)
8. Before opening the polling station at the election day, DRE computers are initialized, tested, and it is ensured that the electronic ballot box of the DRE computer is empty.
9. Voter access cards,(typically used in Belgium) handed out by election officials only for eligible voters, guarantee that a voter has one vote only. The card allows the voter to cast his/her vote.
10. The voter access card acts as a token given a one-time access to the DRE computer. It does not carry other information which could be used to link the vote to the voter.
11. The voter access card is reinitialized for every new eligible voter.
12. Electoral officers watch themselves that voter access cards are not misused to insert additional ballots into the electronic ballot box.

13. At the end of the election day, DRE computers are stopped from collecting votes, and that they tally the votes correctly. The result is printed on a paper trail and cannot altered afterwards.
14. The totals of all DRE computers are summed up and published correctly.

This lists presents a detailed list of trust assumptions that are not necessary TC-dependent, but rather generic, and reflect the “ideal” trust model.

An analysis done by Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten [Feldman et al.] for a particular DRE computer demonstrated that the above presented trust model is by far idealized. They showed a number of manipulations of the DRE computer:

- They manipulated the DRE computer’s software such that the result obtained at the end of the election period would not have expressed the voters’ opinion (violation of trust assumptions 4 and 14).
- They opened the mechanical lock to replace the memory module to load the manipulated software (violation of trust assumption 5).
- They demonstrated design flaws by showing that the DRE computer’s software update mechanism can be misused to spread malicious code from one DRE computer to another (violation of trust assumptions 4, 6, and 8).

Earl Barr, Matt Bishop, and Mark Gondree [Barr et al.] reported the lack of threat models and system models (trust assumption 1 from above) for DRE computers. They argue that without such models, voting standards cannot ensure the integrity and accuracy of the voting process when e-voting systems are involved.

Kiosk voting machines impose further trust assumptions in addition to the ones defined for DRE computers.

1. A vote being entered by a voter must be sealed by the kiosk voting machine prior to transmission such that it is not possible to alter it.
2. A vote being entered by a voter must be sealed by the kiosk voting machine prior to transmission such that it is not possible to read it.
3. A vote being transmitted to the remote vote server cannot be lost or duplicated.
4. A vote is entered into the electronic ballot box and cannot be read by any one prior to the termination of the election period.
5. A vote can be transmitted with a valid voter access card only.

Trust assumptions must be made for the central voting server, too. Since the central voting computer is similar to the one being used with Internet voting, the same trust assumptions hold true.

5.1.3 Supporting technologies

Breaking down the trust model into detailed and specific trust assumptions as in 5.1.2 helps envisioning how TC technologies and general trusted infrastructure concepts can fulfil each of these assumptions. In the following, we show refer to the assumptions in 5.1.2 by indicating how Trusted Infrastructure concepts can fulfil some of these assumptions.

First, it is important to note that the specifications of architecture designs for trusted infrastructures focus on the mechanisms, protocols and components on which the trust model relies. This means that in principle, the specifications of the trusted computing base (TCB) should be enough to guarantee the basic security requirements, and at least part of the behavioural requirements. Therefore, a DRE or a kiosk voting machine with a rigorously delimited and specified TCB should be able to guarantee a high level of trustworthiness. This represents a step forward towards addressing assumption (1) on rigorous specification of the machine. However, it is self-evident that the more sophisticated the TCB is in terms of security techniques, the higher the level of security expertise will be required for evaluation. This makes assumption (2) on specification reviewing considerably harder to fulfil, but still achievable. Nevertheless, security experts representing governmental bodies should be responsible for verifying that the actual implementation of the TCB architecture complies with the reviewed specification. For this purpose, manual checking as well as sophisticated tools are generally used for this type of verification. However, it is indispensable to make the TCB as small as possible in order to guarantee a reliable verification. In [Sastry], a TCB architecture for voting machines is proposed in a way to reduce the size of the TCB code as much as possible without compromising on the critical security functionalities to be verified. Assuming that the “vote selection code” in a DRE or kiosk machine is the most complex in the system, the focus is directed towards the “confirmation process” which is considerably easier to verify. Moreover, its analysis allows verifying many security properties without the overhead of verifying the “vote selection code”. Therefore, the proposed architecture abstracts the “vote confirmation code” into a modular component which can be verified for its compliance with certain specifications as part of the overall TCB specifications. This leads to a substantially reduced TCB size, with the vote confirmation logic being the only aspect to require a rigorous code review by security experts. If these guidelines (process isolation, small TCB) on TCB architecture are adopted [D3.9, 3.2.3], trusted infrastructures would be advantageous in terms of fulfilling assumption (3) on verifying the implementation of DREs and kiosk machines against standard specifications, guaranteeing the needed reliability of the trusted components.

From that point on, TC functionalities (based on hardware anchors, such as TPM) can guarantee the integrity of the TCB, and help turning down malicious (or revealing any accidental) attempts for tempering with any hardware or software components which are part of the TCB. This is a fundamental advantage of TC technologies. As explained in [D3.9, 3.2.1], a verifiable authenticated and secure boot is an essential aspect of TC-supported machines. Therefore, assumption (4) can be effectively addressed if adequate TC techniques (TPM, Integrity measurements, authenticated boot) are integrated in the TCB.

Certifying the DRE or kiosk machines could require a complicated process. However, Common Criteria or ITSEC standards can be used here for solely certifying the TCB or parts of it (e.g. the security kernel) according to certain evaluation criteria or level (e.g. CC EAL 1-7). This would allow assumption (6) to be addressed as far as security requirements are concerned: a certification can be issued for a DRE machine, stating which particular components meet the criteria of a certain level (e.g. EAL5 in case CC is chosen as a standard).

Assumption (8) can also be addressed by means of TC functionalities. In fact, the Trusted Boot and Remote Attestation features [D3.9, 4.2.3] can be used in this context in order to 1) perform a sequence of integrity measurements of code running on the DRE (which is based on the “Chain of Trust” concept) and 2) report the end result of these measurements by means of standardized attestation protocols to a central server. This server will be able to verify the current status of software stack running in the DRE (including, e.g., the state of the electronic ballot box) and certify that the DRE is in an acceptable state to start operation. This process can take place just before the start of the voting, i.e. during initialization of the machine.

The different aspects of dealing with voting cards might fall inside or outside the scope of trusted infrastructures and their benefits. In fact, this issue highly depends on the overall trust model which includes election officials, personnel, etc… All play a role in the election process. The assumptions listed in the previous section imply that these people might themselves not be trusted (otherwise, there would be no need for assumptions (4, 5 and 7) since everyone other than voters would be trusted not to manipulate the system). If these officials are trusted, there should be no need for additional security requirements to account for assumptions (9, 10, 11, and 12). The functional requirement of making a voting card enable a single vote in the system would be enough. However, if the trust assumptions should be fulfilled by technical means, additional security requirements are needed.

For example, if an election official at the polling station is not trustworthy, he might exchange the voting card with another one that allows more than one vote after each re-initialization. This would allow voters who are aware of that to cast more than one vote, i.e. assumption (9) would fail unless assumption (12) is satisfied. This is an example of administrative security requirements or assumptions that need to be fulfilled, and that can not be covered by Trusted Infrastructures.

The additional requirements entailed by Kiosk voting machines can be addressed by a combination of TC technologies and secure protocols. “Trusted Channels”, a basic concept of Trusted Infrastructures can be used for this purpose. They are secure channels that require mutual attestation of trustworthy configuration of endpoints prior to establishment. In [Gasmi et al.], a design and implementation of a Trusted Channel based on a combination of TC functionalities and TLS protocol were presented. Such channels can be established between the Kiosk machine and the voting server to address additional assumptions (2) and (3) in order to avoid reading, omitting or duplicating votes during transmission. Additional assumption (1) can be fulfilled by using the TPM sealing functionality [D3.9, 3.2.2] for binding the votes to the kiosk machine identity and configuration and thus sealing the votes which are stored on its hard disk.

We should note here that assumption (10) on non-identifiability of the voter through the voting card implies that the identification and authorization of the voter to vote is done by election officials (at a first stage), and falls therefore outside the scope of Trusted Infrastructures.

So as we see, Trusted Infrastructure concepts can help satisfy many of the assumptions of the trust model for DRE and Kiosk machine-voting. However, many other administrative assumptions need to be met, and fall outside the scope of Trusted Infrastructures.

A note on smartcards

Usually, a smartcard (which is assumed to be the form of identification cards) is trusted to preserve data or keys that play the role of authentication credentials for the smartcard holder to a certain system. A PIN that is only known to the smartcard holder enables usage of the smartcard. Therefore, the combination of smartcard and PIN allows successful identification and authentication of the smartcard holder. This trust model might not be suitable to address the assumptions (9, 10, 11, and 12). In fact, it assumes that the smartcard holder does not have interest in passing his smartcard and PIN to another person, which is not valid in an e-voting scenario, because vote coercion and trade cannot be avoided.

5.2 Internet voting

5.2.1 Description

Remote e-voting, refers to the casting of ballots at private sites (home, office, school, …). If the transport of the ballot is transmitted via the Internet, then this type of system is often called *i-voting system*. Typically, personal PC machines, laptop machines, or Personal Digital Assistant machines are used for casting the ballot. The eligibility of a person to vote must be part of the so-called *voting protocol*. This implies that the voting protocol must include a means to verify the eligibility of the voter, as well as a means to cast the ballot such that the integrity and privacy of the vote can be guaranteed. Common to the type of machines used by the voter is the fact that they are *not* administered and operated by election officials since they are proprietary voting devices, and that they are open software systems[4].

A variant of this type of i-voting is the provision of PC machines at polling stations. These machines are connected to the Internet, too, but prepared and operated locally by election officials. Yet another type of remote i-voting exists where computers are also deployed and operated by election officials, but the traditional safeguards cannot be guaranteed since computers at sites that are publicly accessible. Typical sites are shopping malls, post offices, libraries, and schools. Since the latter types of e-voting computers resemble the *kiosk voting* its underlying trust assumption are similar. The only (important!) difference occurs for the determination of the eligibility of a voter: For this part, the same trust assumptions must be made that exists for i-voting.

To set up the trust assumptions for a broad range of i-voting systems let’s oversimplify the setup of computers: On one hand, there are the personal PC machines, laptop machines, or Personal Digital Assistant machines at the voters’ realm. On the other hand there is the central voting infrastructure consisting of one or more voting servers. We list trust assumption for both parts of the voting system as well as for the communication among them.

It is also assumed that an eligible voter has received or knows reliably his/her secret authentication data in order to prove the eligibility to vote at the e-voting system, and did not forward this secret data to anyone else..

[4] Open software systems are computers that allow the user to add, remove, or modify software that is installed on the computer.

5.2.2 Voter device

In the following, we give the general trust assumptions that underlie the operation of the device used for voting, and then shed the light on how Trusted Infrastructures can help satisfy these assumptions.

5.2.2.1 Trust Assumptions

For the computers for i-voting at the voters' realm, trust is based on the following assumptions:

1. The computer's hardware is working correctly. Especially, the central processing unit, the memory, any auxiliary hardware component such as the floating point accelerator, the memory mapping unit, cryptographic devices (if any), the network cards, the peripheral devices such as keyboard, mouse and display, disk, and USB bus are all working as expected.
2. The computer's software is working correctly. No malicious software such as viruses or spyware was added as a side effect upon connecting the computer to the Internet, or loading any kind of data from storage subsystems such as CD-ROM, magnetic tape, secondary disk, USB memory stick, etc.
3. Protection software such as firewall and spyware and phishing protection utilities are in operation and properly configured.
4. The voting software either installed from a storage device such as CD-ROM or USB memory stick or received from an Internet site prior a vote is trustworthy. There exists a means to check the trustworthiness of the voting software.
5. Alternatively, the software such as a browser, a Java applet, and/or an active control being used while the voter is performing his/her vote is trustworthy. There exists a means to check the trustworthiness.
6. The voter's computer is connected to the central voting server upon performing the voter's authentication. The voter has a means to check the authenticity of the central voting server.
7. The voter's authentication data reliably identifies the voter's right to vote only once.
8. The voter's authentication data reliably prevents the voter to vote more than once.
9. The voter receives from the central voting server an authentic ballot. There exists a means to check the ballot's trustworthiness.
10. The voter's computer does not alter the voting options given by the voter. The voting options are put onto the electronic ballot as entered by the voter.
11. The voter's computer is connected to the central voting server upon casting the voter's ballot. The voter has a means to check the authenticity of the central voting server.
12. The voter receives from the central voting server an authentic confirmation that the central voting server has received the electronic ballot. The confirmation data does not contain any information that allows the voter to proof the way he/she voted.

5.2.2.2 Supporting Technologies

Remote internet voting clients are general-purpose, open-software computers. As such, it is possible to install arbitrary application and system software, including low-level software components such as drivers for controlling and managing hardware components such as keyboard, display, and the network card. Thus, the users of these computers are exposed to additional risks upon casting their votes:

- In a home setting, so-called spyware can compromise the secrecy of the electronic ballot. Malicious code could have been added to the computer which allows an adversary to explore the voter's voting choice without being noticed by the voter.
- Similarly, if the computer is operated in an institutional setting, administrators can install to monitor the user's activities. If used properly, these tools can be of great value for the user and the institution. In the context of remote i-voting, however, these tools compromise the secrecy and privacy of the voter.
- Computer viruses and malware can be deployed on the voter's computer such that this type of malicious code changes the vote options given by the voter prior to safeguarding and transmitting it. The voter believes that the ballot expresses his/her vote options since he/she cannot tell it from the information given on the display.

Ronald L. Rivest has coined the term *secure platform problem* [Rivest2] for this type of problem. It is widely believed that the secure platform problem is the Achilles heel of remote i-voting system.

Therefore, Trusted Infrastructures might be able to establish the necessary trust in the voting devices in several ways.

In fact, there is an essential requirement of reliability that might be problematic if voters want to use their proprietary devices. This requirement is irrelevant in the case of machine voting since the voting machines are provided by the state. The eligibility for voting, which is usually checked through authentication, is not enough since a voter would additionally be responsible for the correct operation of many hardware and software components of his device. Therefore, a considerable part of satisfying assumption (1) falls outside the capabilities of Trusted Infrastructures. Nevertheless, some security-critical software components must not only be checked for reliability, but also for trustworthiness. In fact, as opposed to the machine-voting scheme, the proprietary devices of voters are expected to include different operating systems and software components, which make the device vulnerable to several kinds of attacks. These attacks could be mounted, e.g., by software providers who can include backdoors in their software to manipulate the vote, Trojan horses or malware on the device, etc… Other attacks could be mounted by voters who might attempt to manipulate the voting software in order to, e.g., cast more than one vote.

However, special client device architecture have been conceived and implemented in the scope of Trusted Infrastructure in order to address this problem. Trusted client endpoints are a hot subject that has acquired considerable attention in the scientific research and development arena [Tura09, Tnc06]. They rely on the same TC functionalities used for machine voting (see 5.1.3), namely integrity measurements and remote attestation. However, the typical architecture for client endpoint security allows running other legacy OSes and software components on the same client device. For example, Turaya Security Kernel [Tura09] which leverages virtualization and TC techniques provides a suitable architecture for trustworthy proprietary client devices. It can be installed on the voter's computing device which would allow him to run several operating systems concurrently on top of it, and in separate virtual environments (compartments). These virtual environments (also called virtual machines) all share the same hardware components. In this case, the voter would have a TCB on his device, on top of which run his own native applications in several untrusted compartments. Only one compartment running on the security kernel will have to be trusted, and that is a compartment dedicated for i-voting.

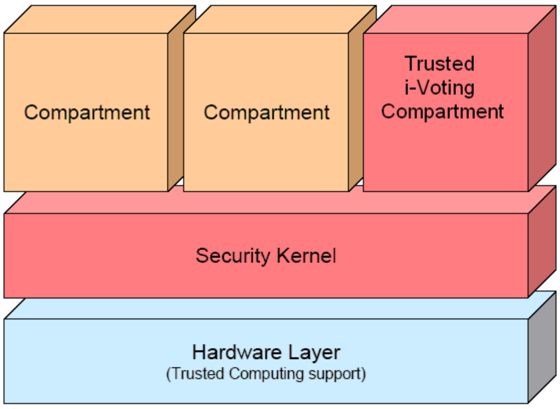

Fig.1: Security Architecture for i-Voting proprietary devices.

The trusted i-voting compartment would run a minimal operating system with only a web browser that allows connection to the i-voting server. The trusted compartment can be configured in a way to allow only the web browser (or voting software) to appear when the voter switches the focus to the compartment, with the exclusive capability to access the i-voting server via a trusted channel. Integrity measurements of this compartment can reflect this trustworthy configuration, and a remote attestation process would be required to attest the measurements of the TCB as well as the i-voting compartment to the i-voting server. The latter would be able to verify the trustworthiness of these relevant components against the i-voting security policy. If this is verified, the connection is established, and the voter would be able to vote. This type of architecture can effectively fulfill assumptions (2, 3, 4, 5 and 10). It has already been proposed in [Alkassar et. al], with detailed cryptographic protocols on establishing the trusted channel between the voter's device and the i-voting server based on TC keys.

Assumptions (6) and (11) can be addressed by means of public certificates used within a trusted channel scheme. Certificates of voting servers as well as so-called TPM attestation certificates[5] can be used to authenticated/authorize both parties (i.e voter's device and i-voting server) before establishing a channel.

[5] A TPM generated certificate that includes fingerprints of TCB and trust compartment configuration.

Assumptions (7) and (8) on the voter's authentication credentials (indicating eligibility for voting) fall outside the scope of a Trusted Infrastructure for i-voting since these credentials do not represent "trusted" components. Nevertheless, there are many reliable authentication credentials that can be used, such as a public certificate that can be provided offline to each eligible voter. However, the problem would be to make sure that this certificate (or any other sort of credentials) is only accessible to the i-voting compartment. This can be achieved by means of TC sealing functionality. The certificate would be encrypted with a "TPM binding key". This key would only be accessible to the trusted i-voting compartment. Therefore, it would be impossible to decrypt the certificate outside of the i-voting compartment, which would prevent its leakage to other vulnerable compartments subject to attacks.

Assumption (9) and (12) require another additional security requirement which is a "trusted GUI". This is required in order to guarantee that the ballot received from the server upon voting is authentic. In fact, without a trusted GUI, an attacker can manipulate the GUI, which would allow him to show another ballot on the screen in order to fake the result.

5.2.3 I-Voting Central Server

Again, we give the general trust model underlying the operation of the i-voting central server, and explain how Trusted Infrastructures can help fulfil the trust assumptions.

5.2.3.1 Trust Assumptions

Next, trust is investigated for the central voting infrastructure. We assume that the voters' register is set up consistently and correctly, and that voters' credential data, if any, is generated, safely protected, and communicated to the eligible voters via a safe, independent communication channel

For the central voting infrastructure trust is based on the following assumptions:

1. There exists a clear and precise definition of the hardware and software components in operation for the central voting infrastructure.
2. The hardware and software components and their interworking have been tested thoroughly.
3. The deployment and operation of the central voting infrastructure has been approved by an independent governmental or otherwise trustworthy body.
4. Involve hardware and software components cannot be changed without appropriate approval from the independent governmental or otherwise trustworthy body while an election or poll is prepared, conducted, terminated, and archived.
5. The central voting infrastructure is physically well protected. Hardware is installed at safe places. Physical access is limited.
6. The central voting infrastructure is protected from any threat coming from the Internet via firewalls or other appropriate software.
7. Access to the infrastructure while being in operation shall be subject to formal operation procedure. Changes shall be logged and approved.

8. The generation of public and private key pairs is conducted under observation of the independent governmental or otherwise trustworthy body. Private keys are safely stored. Public keys are published appropriately, for example by enclosing them in signed certificates.
9. At the beginning of the election or referendum, it is ensured that the electronic ballot box of the central voting infrastructure is empty.
10. Given the credential data of a voter, the central voting system trustfully checks the eligibility to vote of the voter, and safely returns the according indication.
11. Every received electronic ballot is sealed such that its content cannot be altered.
12. Every received and valid electronic ballot is entered into the electronic ballot box.
13. Every received electronic ballot is sealed such that its content cannot be read before the closing of the electronic ballot box.
14. It is impossible to insert an electronic ballot into the ballot box which does not originate from an eligible voter.
15. It is impossible to insert more than one electronic ballot into the ballot box received by an eligible voter.
16. It is impossible to link an electronic ballot being received to the voter or to the computer that was used by the voter.
17. At the end of the election period, the central voting infrastructure is stopped from collecting votes.
18. At the end of the election period, the electronic ballot box is sealed.
19. At the end of the election period, the sealed electronic ballot box is archived for potential recounting.
20. At the end of the election period, log and audit records are sealed and archived.
21. The central voting infrastructure tallies the votes correctly. The result is sealed, published, and archived.

Trust on the communication between the voter’s computer and the central voting infrastructure is based on the following assumptions:

1. The communication is based on a reliable channel between the sender and the receiver. Messages occur in order, and are not duplicated. Messages cannot be altered, and cannot be read by intermediate agents or any third party.
2. It is not possible for an intermediate agent to insert a message into the channel associated between the sender and the receiver.
3. The voter’s electronic ballot is sent via a so-called anonymous channel. It is impossible to trace the sender’s computer upon having received the ballot at the receiver’s side.

5.2.3.2 Supporting Technologies

Many of the assumptions underlying a secure and reliable operation of a central i-voting server can be addressed by functionalities mentioned above. To avoid repetition, we can say that assumptions (3, 4, 5, 7, 8, 9, 17, 18, 19, 20, and 21) are mostly of administrative nature, and therefore fall outside the coverage of Trusted Infrastructures, unless it is decided to automate certain processes instead of giving the governance responsibility to election officials. Assumptions (1) and (3) can better be accounted for in a Trusted Infrastructure since the core security functionalities are part of the TCB, which should be small and easier to verify as explained in 5.1.3. However, it should be noted that a TCB can account for security aspects, but other components whose reliability is critical must also be checked.

Assumption (6) is typically addressed with a system architecture leveraging virtualization and trusted computing (see above), which would allow running the i-voting central software in an isolated virtual environment verifiable for its trustworthiness. The trusted channels established with voter's devices can be bound - via TC functionalities - to the trusted i-voting compartment, so as to make the communication between both endpoints only possible between their corresponding trusted compartments. This would guarantee the trustworthy processing of the vote since its casting, throughout its transfer, and until its storage on the central i-voting server.

Assumptions (10-15) which underlie functional security requirements can in principle be fulfilled by an i-voting software that runs inside the i-voting compartment at the server, and whose integrity is measured and verified by the TPM via a trusted boot process. The communication assumptions (1) and (2) could be achieved by the established trusted channel.

5.2.4 Authentication, Authorization and Anonymity of the Vote

We should consider assumptions (16) and communication assumption (3) with a closer look. They both address the non-linkability aspect between the voter's identity (or his device's identity) and the vote. This is a crucial requirement for the anonymity of the vote. In fact, when TC remote attestation is used to attest measurements of the voter's device to the i-voting server, the attestation certificate might include identification information of the TPM, and hence of the device. This would help linking the vote to the device used for voting, and eventually to the voter. In order to address the general anonymity problem in remote attestation, the TCG has specified the *Direct Anonymous Attestation* (DAA) protocol [D3.9, 8.6.3]. This protocol does not involve a trusted third party, and allows TPM certificate not to reveal any identification information. Therefore, it seems indispensable to use the DAA feature of TPM v.1.2 for the sake of non-linkability between device identity and vote assumption (16).

However, the problem still persists since the voter's credentials, which must be supplied for authorization to vote, should contain authentication information in order for the server to identify the citizens who have already voted. This is in fact that the conflict of requirements that has been pointed out in 3.5 between anonymity and eligibility.

One approach to address this problem based on Trusted Infrastructure concepts is to try to isolate the voting process from the authorization process, and to control the communication between these processes. Then, the i-voting software responsible for both processes on the server would be separately verified for their integrity. The i-voting server would have an architecture similar to the one for client devices depicted in Fig.1, but with two trusted compartments, one for authentication/authorization, and another one for voting can be configured. The i-voting client software – which should typically include two steps, one for authentication and one for voting – establishes a trusted channel with each of the two server compartments. The sequence would be as follows:

1. The voter uses his i-voting client software to send its credentials via the first trusted channel established with the authentication compartment.
2. If authenticated (i.e. the authentication credentials are valid and the voter did not vote yet) the authentication compartment returns some *anonymous one-time authorization credentials* to the voter's i-voting client software.
3. The voting device establishes a second trusted channel based on these authorization credentials with the voting compartment of the server, and the voter can cast his vote.

Through this scheme, the voter can be reliably authenticated, but his supplied authentication credentials are processed and stored in an isolated compartment on the i-voting server. On the other hand, his vote is performed using anonymous credentials, and is also processed in an isolated compartment. Therefore, linking the voter identity to his vote is effectively prevented. Of course, it is possible to separate the authentication software and voting software into two physically isolated servers, but this would require two mutual remote attestation steps for the voting device, prior to authentication and voting.

5.3 SMS voting

5.3.1 Description

Using a mobile phone via the GSM network for legally binding elections or votes introduces another kind of remote e-voting. With *SMS-voting*, the voter receives a personalized voting card. Among others, the personalized voting card contains a reference number (used for the identification of the voter during the authentication step), an election event number, a secret code, and a code sheet. The voter enters the reference number for his/her identification, the election event number, and the codes from the code sheet for his/her votes. Basically, codes are random numbers. Then, the data is cast via an SMS message to a voting server. The system replies by asking the voter to enter a PIN (separate from the identification code) and perhaps additional identification codes in a second SMS message. The second SMS message then terminates the voting phase, perhaps by replying with a confirmation message.

Code sheets have been proposed by David Chaum [Chaum].

There are variations in the way code sheets, together with other personalized date, can be used. The above description is the so-called "code number-only" implementation.

5.3.2 Trust Assumptions

For *SMS-voting*, trust is based on the following assumptions:

1. Secret codes are generated randomly for each candidate and for each voter.

2. For each voter, a secret random reference number and a secret random PIN are generated.
3. The reference number, the PIN, and the codes are secretly stored in a database. (The same information is printed on the personalized voting card.)
4. The personalized voting card containing is securely sent to the voter via postal mail.
5. The correct reference number is only used to associate the voter's codes to the voter's vote options; the association is retrieved from the database.
6. An incorrect reference number imply that the received codes are discarded.
7. The voter's vote options are encrypted, thus forming the encrypted ballot.
8. If the PIN is correct then the encrypted ballot is cast to the electronic ballot box. The database entry is marked accordingly.
9. If the PIN is wrong then the encrypted ballot is discarded.
10. Unmarked database entries cannot be misused to cast a ballot.

5.3.3 Supporting Technologies

In this scheme, there are many assumptions that can not be accounted for by implementing Trusted Infrastructures. In fact, the voter's personalized card and corresponding PIN can end up falling in the hands of another person, whether willingly or by repression. It is hard to guarantee trustworthy identification.

As far as anonymity of the vote is concerned (i.e. non-linkability between voter's identity and his vote), it is dependent on the linkability between the reference number and the identity of the voter. This is an administrative matter because the generation of a unique reference number (and corresponding PIN) for each eligible voter takes place ahead of the election, which means that half of the authentication process is done administratively. If this administrative stage is done in a way that does not misuse any correlation between the reference number and the identity of the voter during the election process, then this would help the election process to be anonymous.

The problem persists with the mobile number. The SMS sent by a certain voter indicates the mobile number from which it has been sent. Linking the mobile number to the vote indicated inside the SMS would violate the anonymity requirements. Since the SMS is processed by a single software at the server side, it is indispensable to have a rigorously specified software at the server side, with the ability of the voters to verify the integrity of this software.

Code Voting [Joaquim et al., Helbach et al.] has been proposed to protect elections integrity, but it is fair to say that the underlying cryptographic protocols can be further enhanced by support of TC techniques such as integrity measurements and remote attestation. Specifically in the SMS voting scheme where mobile number and vote decision are transmitted in a single SMS, the exact behavior of the voting software at the server side is crucial and should be measured and verified against trustworthy values. Nevertheless, the same kinds of attacks – and eventually prevention techniques – are valid for mobile phone devices, as is the case with internet voting.

Moreover, if the *receipt-freeness* requirement is considered, the SMS scheme fails short from achieving it since the SMS including the vote is usually stored on the mobile device, and indicates the receiver (voting service), the reference number (which the voter can prove to be correlated to his identity by showing the corresponding PIN), and the vote itself. This stored SMS can be used by the voter to prove his voting decision in the case of repression. A security architecture such as the one in 5.2.2.2 might not be easily applicable on mobile devices since the no particular voting software is used, except for the normal messaging service. Since this service is mobile-dependent, it is much harder to impose any integrity verification of the underlying messaging software. Trusted Infrastructure concepts can be more effective in the normal PC scenario.

5.4 A Note on Legal and Social Aspects

It has become widely noticed that modern lifestyles have helped change the acceptance of modern technologies by citizens to support a voting process. In fact, the inconvenience of getting to the polling station, or the non-availability of the voter in the country on the election day, considerably affect the election turn out, as indicated by surveys conducted by electoral commissions. According to [HMUK], 10.8% of the voters in the local election in Swindon chose to cast their votes via the Internet from home, local libraries, etc… 80% of the those who voted via Internet indicated that this approach has made voting more convenient and accessible.

Surveys by private institutions also reveal that Internet voting will encourage people to vote. A survey by KMPG[6] on the British elections indicated that 57% of people with Internet access would be willing to vote online.

All these evidences prove that *Internet Voting* will most probably be a widely adopted scheme for elections in the years to come.

As far as trust in the e-voting system is concerned, it seems likely to be gradual. Voter's trust will likely come through experience [HMUK]. However, this trust can only be established if the governments are able to provide evidences to voters that the e-voting system specifications makes it robust against fraud and attacks. In fact, voters usually tend to trust election officials, administrators and clerks to be impartial. For example, the *anonymity vs. eligibility conflict* that we addressed in the previous section can effectively be prevented in traditional voting systems. This is because election officials are trusted not to perform correlation between identities of voters and their corresponding votes (impracticality, governmental audit, etc…).

In the e-voting scenario, an erroneous system can be more efficient in making and storing such correlations. Therefore, any forms of e-voting, whether through internet on mobile devices, internet in polling stations, or SMS voting, would require legal grounds and procedural safeguards in order to ensure and assure the voters of the equivalent protection.

[6] Is Britain on course for 2005?' – www.kpmgconsulting.co.uk/research/reports/ps_egov02.html

Particularly, anonymity of the vote, receipt freeness, and robustness of client devices against attacks are the most critical features to be addressed technically and legally by the election bodies, so as to foster the trust of the voters in e-voting schemes.

According to [Oostveen et al.], public confidence in the reliability and adherence of e-voting systems to traditional voting standards and procedures is fundamental to the legitimacy of the electoral process. "Internet voting is likely to lead to changes in how the public maintains confidence in the integrity of elections. Internet systems pose a problem in that the tallying process is not transparent". According to [IPinst], this public confidence relies on the trust in technical experts – as opposed to election officials in traditional elections. Therefore, the trustworthiness of the voting software used at the server is of great importance to voters. Hence, deployment of Trusted Infrastructures for an e-voting scheme could be helpful in this regard if the trustworthiness of critical aspects of the system can easily be evaluated by normal voters. [Oostveen et al.] suggests that only free software can provide the needed transparency. This would make the voting software publicly available and subject to examination by other public experts. However, the trustworthiness of the software and its adherence to the claimed standards and modus operandi should also be verifiable by the voters themselves on the election day. This can be done by certifying the software that has been inspected and agreed on by an independent certification body, and letting the voters as well as the voting administrators verify each other's software for compliance with the certification. Public certificates can be effectively used to this purpose.

To summarize the requirements of a socially acceptable e-voting system, we can say:

1. Specifications and standards of voting software used by both client devices and voting server should be publicly available for inspection and examination by researchers and experts in the security field.
2. The standards to which the software are claimed to adhere should account for the prevention from fraud, attacks and privacy violation at least to the same level of traditional ballot voting schemes.
3. The compliance of the software to the specified standards should be certified security experts of an independent body.
4. A Trusted Infrastructure should be deployed in order to allow both the voters and the voting server administrators to verify each other's software for compliance with the certification.

In any case, it is fair to say that remote voting poses some concerns on the receiver validity, although this preserves the *anonymity* requirement. There is no guarantee that the receiver is actually the person legally enabled to vote, and changes in postal addresses or a possible postal error can lead to an unauthorized person's vote. This can raise legal issues as well, if we consider that the right of a person to vote, using this method, depends on the receipt or not of the letter containing the code. In the case of a mistake or failure to deliver the letter, the person has the right to prosecute to the court.

Contact with the administration by phone or post, is again not assured in case of technical difficulties placing the voter in a problematic position. Character confusion of the personal code given to all citizens can lead to mismatching and therefore the exclusion of a citizen from voting.

The blockage of the system and the fraudulent use of the random code used for voting, could obstruct the vote of a legally enabled person. Again this can raise a legal issue, considering that the person is willing to vote, but due to technical difficulties or faults, he/she is not able to. However, this is not a straightforward issue as the responsibility could be on the government – as the organizer of the elections – or on the vendor of the voting system.

The structure of the application used in the remote voting procedure, leads the voter to think in terms of parties rather than candidates. This can create an issue of misleading – in some sense – the voter to choose without giving a clear presentation of the parties and the candidates of each party.

The voter should have basically three options while voting: to vote legibly, to cast a blank vote, or to cast a null vote. Blank votes are not such a large problem as they are basically the same as the normal vote, but with an empty selection [Esteve et al., 2003]. However the null vote which is a very popular way to vote cannot be limited to cases of error.

Finally, the receipt procedure should be guaranteed to be safe to add more reliability to the whole procedure. However, this is not really feasible as the procedure depends on internal computer protocols.

Touch-screen machine's position is very important as it may lead to an increase of the blank votes [Esteve et al., 2003].

The paper electronic ballot, of which there are different variations [Esteve et al., 2003, Kiayias, 2007], can result to manual counting in case of wrong voting. This will increase the risk for reliability, and also removes the benefit of the electronic procedure.

E-voting is important for disabled or ill voters as mentioned in [Loide et al., 2007], which increases the turnout for voting and makes voting an easier process for this category of voters. However, this can raise concerns on the equipment and the cost needed in order to vote electronically thus leading to possible digital divide issue.

Concluding, e-voting new technologies are not a panacea for democratic process problems like abstention, digital gap and other issues that emerge.

6 Analysis

In this chapter, we give a brief analysis of the general relevance and implications of deploying e-voting schemes with support of Trusted Infrastructures, and mention few technical alternatives to the voting schemes discussed in the previous chapter.

6.1 Relevance and Implications

6.1.1 Technical perspective

The description in chapter 5 shows that Trusted Infrastructures seem to potentially have a substantial effect on the different e-voting schemes. Many Trusted Infrastructure concepts described in chapter 4 can be used in various ways to fulfil the trust assumptions underlying a reliable, trustworthy and secure e-voting scheme. Some of these concepts (e.g. small and verifiable TCB, TPM, integrity measurements) are relevant and apply similarly to the three discussed kinds of e-voting, i.e., machine voting, internet voting and SMS voting. Other concepts, e.g. combination of virtualization and TC techniques such as remote attestation and trusted channels can be used selectively, in different manners and for different purposes according to each of the three schemes.

One observation that can be made is that these concepts can only be applied if certain hardware and software properties are assumed. For example, whether the software being verified for trustworthiness is on the voting server side or the voter client side, this software is assumed to be small, non-complex, verifiable and modular. Otherwise, it would hard, if possible at all, to verify the compliance of this piece of software to certain specifications and standards.

Another observation is that the deployment of a Trusted Infrastructure for remote e-voting requires specific hardware trust anchors (TPM) and specific voting software to be available on the platforms used during voting. This implies that these hardware and software components should be available on each voter’s device in order to ensure availability of the service to each citizen. If this is achieved, then the voting can be performed easily enough by the normal people (i.e. with low computer literacy).

It should also be noted that integrating Trusted Infrastructure concepts in an e-voting system is intended to provide a certain level of security which is roughly equivalent to the security of traditional voting schemes that is usually ensured by trusted election officials. In fact, relevance of the techniques discussed in chapter 5 is evaluated according to the trust assumptions which are usually fulfilled by impartial election officials and governmental auditing. This means that Trusted Infrastructures would merely help the e-voting system compensate for the insufficient trust in election officials due to the use of technology, and should not be expected to add more security to the voting process in general.

6.1.2 Social and Legal perspective

Trusted Infrastructures are only relevant in a comprehensive scheme that allows security experts to evaluate the voting software, and ensure that it can be measured for its integrity.

The awareness of the voters about the value of the certification of the e-voting machines or i-voting server and client software is essential.

The deployment of Trusted Infrastructure for an e-voting system requires new legal grounds that need to be conceived and adopted in a way to account for the new measures of liability and accountability of the people and institutions involved in the voting process. In fact, security experts, governmental and election officials, certification institutions, as well as voters themselves hold responsibilities which are slightly different from the ones assigned in the scope of traditional voting schemes. This implies that accountability of any of the people involved in the process is subject to new measures and different evaluation perspectives, and the legal rules should take that into consideration.

The advantage of integrating Trusted Infrastructure for e-voting lies in the confidence it gives to voters that state-of-the-art security technologies that are generally dedicated for achieving trust in system software (even remotely) are integrated in the e-voting system. This helps achieving a certain level of assurance in the voters' community with regard to the equivalence – in terms of trustworthiness – between the e-voting system and traditional voting schemes. If this assurance is reached, the sole fact that e-voting can make the process more convenient and accessible would lead to a gradual acceptance of the concept among citizens. This in turn would eventually improve the voting turn out.

Nevertheless, as e-voting primarily lies on TC technologies, it requires to maintain democracy, privacy, verifiability and fairness features, thus implying accuracy, security and effectiveness of the electoral procedure [Kiayias, 2007]. Consecutively the TC technology that will be used in the electoral procedure should by all means ensure trust, maintain integrity and transparency of the procedure as a whole. In case of failure to meet the above requirements, responsibility should be shifted upon the government or the TC system vendor. Placing responsibility upon such parties can only be achieved through legal framework.

E-voting and i-voting require at first computer possession with proper hardware and software (protection software, browser or voting software) and connection to the central voting server, as mentioned in the previous chapter. These assumptions make the procedure more complex and pose some constraints in the e-voting procedure.

Computer literacy is another issue that should be visited under the light of e-voting, as the voters should be properly educated to use the computer and to be familiar with the voting procedure. Reasonably, older or computer illiterate people will face difficulties in following this method of voting, and this may decrease the voting turnout concerning that age group. On the other hand, because e-voting and i-voting makes voting easier, younger people will possibly vote in a larger scale and therefore it may increase the voting turnout in this age group.

Furthermore, the fact that null votes are not permitted, may increase the valid votes percentage. Blank votes as discussed will continue to exist [Esteve et al., 2003], however this will not influence the voting turnout in any different way than it already does in the normal voting procedure.

6.2 Alternatives

Besides direct-recording electronic (DRE) voting systems and internet voting, hybrid approaches exist that strive to combine the advantages of both, the (traditional) paper-based voting schemes and voting schemes that are exclusively driven by machines. Hybrid schemes, or better known as optical scan voting schemes, consist of a paper ballot based and an electronic part, the first providing the convenience of traditional voting protocols and the latter speeding up the tabulation and tallying processes. The bridge between the traditional ballot and the electronic accumulation is an optical scanner.

The trust in the election result does not depend on the integrity of the voting machinery and the correctness of its software (as it is in case of DREs), but rather on the verification of the election result. Unlike the traditional schemes where the election authorities are responsible for tallying and verifying the result, in most modern optical scan voting systems each voter can verify the election result herself. In order to verify that her vote has been counted, the voter retains a kind of receipt and compares it later to a public list of counted votes. The receipt needs to carry enough information for revealing any manipulation of the ballot after it has been marked by the voter and scanned, but should not reveal enough information for proving the vote itself. Otherwise the voter can be coerced to a specific vote.

6.2.1 Prêt-à-Voter

Chaum's Secret-Ballot Receipts[Chaum2] satisfy these properties. A voting scheme building on Chaum's ideas and simplifying the receipt format has been proposed by Ryan [Peacock et al.] under the name "Prêt-à-Voter". Chaum further enhanced the protocol later. In the Prêt-à-Voter scheme, ballots have two sides, one containing a randomised list of candidate names and the other one providing the space for the vote. Figure 1 shows a hypothetical ballot.

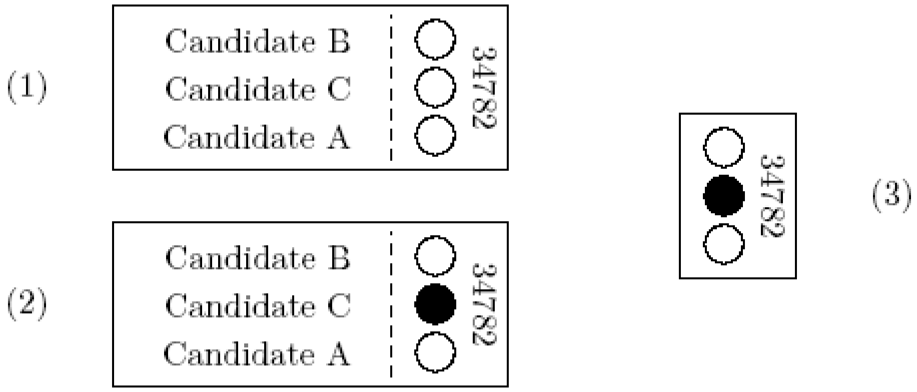

Fig.1: Prêt-à-Voter ballot. The clean ballot (1) is handed out to the voter who marks the area next to the candidate of her choice (2) and cuts off the candidate list (3). The candidate list will be destroyed and the remaining ballot will be scanned, signed, and handed out again as receipt to the voter.

Each ballot carries a pseudo-random ballot number which is printed on the side containing the vote. The other side is shredded after a candidate has been marked and thus any public link between candidate and vote is destroyed. The remaining side of the ballot is scanned, signed, and anonymously published by the election clerks. Afterwards, the remaining side of the ballot can be taken along by the voter as a signed receipt. It can be checked against the list of ballots which is published by the election clerks. If the scanned version of the ballot is wrong, the voter can show his receipt and the valid signature and therefore prove the mistake or fraud. The same applies, if the scanned ballot cannot be found in the list.

In the Prêt-à-Voter scheme, the voter has to trust the election authorities that the ballot numbers are generated with a proper pseudo-random number generator as well as the sequence of candidate names on each ballot. Moreover, the voter has to trust the authorities that they keep the specific sequence of candidates for each ballot secret. Enhancements of Prêt-à-Voter use cryptographic means to relax the latter requirement and make records about sequence of candidates [Ryan et al., Ryan, Xia et al.].

6.2.2 Scantegrity

With Scantegrity [Carback et al.] and Scantegrity II[7] , Chaum et al. proposed a kind of a module which can be added to any existing optical scan voting system and yields end-to-end (E2E) verifiability, i.e., the voter can verify that her vote went into the tally and the tally is not manipulated with a reasonably high probability. In particular, Scantegrity can be used to enhance Prêt-à-Voter schemes with E2E verifiability. The key idea is to use invisible ink (Scantegrity II) to prepare each area for voter's marks. Unlink in common optical scan voting systems, the voter will not fill the mark area with visible ink, but reveal a confirmation code with a special pen. Each vote mark area covers a unique confirmation code, i.e., unique not only on the ballot but in the entire election. This code will later be used by the election authorities to recover and tally the vote. A voter who wants to verify the election has to write down her revealed confirmation code for checking it later against a public list, similar to the Prêt-à-Voter scheme. The link between confirmation code and voter has to be protected by the underlying optical scan voting system.

This scheme already allows using optical scan voting schemes, and enhances them by the E2E verifiability property. It is therefore particularly interesting for elections where optical scan voting systems are already established and installed.

[7] https://www.usenix.org/events/evt08/tech/full_papers/chaum/chaum_html/ScantegrityII.html *last visited: 06/09*

7 Summary

In this deliverable, we laid down the general requirements for E-voting, and emphasized the requirements that are conflicting among each other in some aspects. Specifically, the anonymity of the vote (a fundamental requirement entailing non-linkability between the identity of the voter and his vote) and the eligibility for voting (another fundamental requirement entailing authentication based on identification of the voter) are inherently in conflict. Another case is that of the receipt-freeness requirement (necessary to avoid vote coercion and vote trade) and the individual vote verifiability requirement (necessary to assure the voter that his selection has been tallied correctly).

Trusted Infrastructures concepts, which are explained in chapter 3, include a set of mechanisms, components and protocols that are combined and adequately used to provide a certain level of trust in IT infrastructures. Trusted Computing, a key concept underlying Trusted Infrastructures, is based on the concept of integrity measurements, and relies on a TPM, a tamper-resistant security chip. The TPM integrates the core root of trust in a system, and allows encrypted software fingerprints stored inside it to be communicated via remote attestation protocols to a remote verification party. Based on this concept, trusted channels are designed, which basically are secured channels established between trustworthy endpoints. The features of Trusted Infrastructures motivated the work on studying their relevance to e-voting schemes.

Three different kinds of e-voting schemes were analysed in chapter 5, namely machine-voting, internet-voting, and sms-voting. For each of the three schemes, a description, a trust model and the Trusted Infrastructures supporting concepts are explained with regard to how they can address the trust assumptions of the scheme. This analysis showed that Trusted Infrastructures can make a considerable impact on e-voting systems in various ways to fulfil the trust assumptions underlying a reliable, trustworthy and secure e-voting scheme.

The analysis in chapter 6 showed that some of these Trusted Infrastructure concepts (e.g. small and verifiable TCB, TPM, integrity measurements) are relevant and apply similarly to the three discussed kinds of e-voting. Other concepts, e.g. combination of virtualization and TC techniques such as remote attestation and trusted channels can be used selectively, in different manners and for different purposes according to each of the three schemes. In any case, the deployment of a Trusted Infrastructure for e-voting assumes that voting devices, whether public or proprietary, must include e-voting software with specific characteristics, such as verifiability, non-complexity and modularity. This is necessary to verify the compliance of the software against trustworthy specifications ahead of elections. Moreover, the involved devices must include specific software (e.g. security kernel with virtualization capability) and hardware (TPM chip).

Nevertheless, when deployed for an e-voting system, Trusted Infrastructures should not be expected to add more security to the voting process as compared to traditional voting schemes. The must be deployed however in order to compensate for the insufficient or irrelevant trust in election officials due to the use of technology.

We also summarized the requirements for a socially acceptable e-voting system with Trusted Infrastructure support. They entail rigorous specifications and standards for the e-voting software to be public available for inspection, preventing fraud, attacks, and privacy violation, and certified by independent bodies. Trusted Infrastructures help verifying the compliance to these certifications by the voters and voting server administrators, and can therefore give confidence to each of the parties that the level of trust in the technical infrastructure is equivalent to the trust given to election officials in traditional voting schemes.

We also discussed a couple of alternatives to machine and internet e-voting systems that present a hybrid approach combining traditional (paper-based) approaches with machines, namely the Prêt-à-Voter scheme and the Scantegrity scheme.

8 Bibliography

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

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

[Rivest et al.] Rivest, R. L., Smith W. D.: Three Voting Protocols: ThreeBallot, VAV, and Twin. EVT'07: Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology, Boston, MA (2007).

[Rivest2] Rivest, R. L.: Electronic Voting. Proceedings of Financial Cryptography '01, (February 2001). http://theory.lcs.mit.edu/~rivest/Rivest-ElectronicVoting.pdf.

[Feldman et al.] Feldman, A., Halderman, J., and Felten, E. Security Analysis of the Diebold AccuVote-TS Voting Machine. Center for Information Technology Policy, Princeton University, Princeton, NJ (Sept. 13, 2006); http://itpolicy.princeton.edu/voting/.

[Barr et al.] Barr, E., Bishop, M., Gondree, M.: Fixing E-Voting Standards. Communications of the ACM, Volume 50, Number 3, (March 200).

[Chaum] Chaum, D.: SureVote: Technical Overview. Proceedings of the Workshop on Trustworthy Elections (WOTE '01), presentation slides, August 2001 http://www.vote.caltech.edu/wote01/pdfs/surevote.pdf.

[Carback et al.] D. Chaum, A. Essex, R. Carback, J. Clark, S. Popoveniuc, A. Sherman, and P. Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security & Privacy, 6, 2008.*

[Chaum2] David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2, 2004.

[Ryan] P. Y. A. Ryan. Prˆet `a voter with paillier encryption. *Journal of Mathematical Modelling of Voting Systems and Elections: Theory and Applications*, 2008.

[Peacock et al.] T. Peacock and P. Y. A. Ryan. Prêt-à-Voter: a system perspective. Technical Report CS-TR:929, University of Newcastle, 2005.

[Ryan et al.] P. Y. A. Ryan and S. Schneider. Prˆet `a voter with reencryption mixes. In *Proceedings of the 11th European Symposium on Research in Computer Science (ESORICS'06)*, LNCS, pages 313–326, 2006.

[Xia et al.] Zhe Xia, Steve A. Schneider, James Heather, and Jacques Traor´e. Analysis, improvement and simplification of prˆet `a voter with paillier encryption. In *Proceedings of the conference on Electronic voting technology*. USENIX Association, 2008.

[ArFaSm97] Arbaugh W. A.; Farber D. J.; Smith J. M. (1997) A Secure and Reliable Bootstrap Architecture, Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, May 1997.

[Gasmi, et al.] Y. Gasmi, A.-R. Sadeghi, P. Stewin, M. Unger, and N. Asokan. Beyond Secure Channels, 2007. ACM Workshop on Scalable Trusted Computing 2007.

[Sastry] N. K. Sastry. Verifying Security Properties in Electronic Voting Machines. *PhD Dissertation in Computer Science.* University of California Berkeley, 2000.

[Tura09] Turaya Security Kernel, Sirrix AG,http://www.sirrix.de/media/downloads/54926.pdf

[D3.9] Ammar Alkassar and Rani Husseiki (Eds.). FIDIS deliverable D3.9 "Study on the Impact of Trusted Computing on Identity and Identity Management". FIDIS NoE Consortium – EC Contract No. 507512. 6th Framework Application of European Commission. 2007.

[Tnc06] Interop Labs. What is TCG's Trusted Network Connect? *Network Access Control Interoperability Lab.* 2006

[Alkassar et. al] A. Alkassar, A. Sadeghi, S. Schulz, M. Volkamer. Towards Trustworthy Online Voting, *Fundamenta Informaticae, IOS Press and European Association for Theoretical Computer Science, EATCS,* 2006.

[Joaquim et. al] Rui Joaquim, Carlos Ribeiro. CodeVoting Protection Against Automatic Vote Manipulation in an Uncontrolled Environment. *Vote-ID,* 2007

[Helbach et al.] Jörg Helbach, Jörg Schwenk. Secure Internet Voting with Code Sheets. *Vote-ID,* 2007.

[HMUK] HM Government, UK Online. In the service of democracy. A consultation paper on a policy for electronic democracy. *E-democracy consultation office,* 2002.

[IPinst] Internet Policy Institute (2001) Report of the National Workshop on Internet Voting: Issues and Research Agenda. March 2001.

[Oostveen et al.] A. Oostveen & P. Besselaar. E-voting and media effects, an exploratory study. Department of Social Sciences. *Paper for the EMTEL Conference, London,* April 2003.

[Esteve et al., 2003] Barrat i Esteve J./ Reniu i Vilamala J.-M., *Legal and Social Issues in Electronic Voting-Report on the Catalan Essays during the Elections of November*, 2003, http://www3.unileon.es/dp/aco/area/jordi/treballs/evot/xile.pdf

[Kiayias, 2007] A. Kiayias, et al. (2007). `Tampering with Special Purpose Trusted Computing Devices: A Case Study in Optical Scan E-Voting'. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual,* pp. 30-39.

[Loide et al., 2007] Erik Loide and Ülle Lepp (2007). 'E-voting - a Key to Independence for All.'. In *Conference and Workshop on Assistive Technology for People with Vision and Hearing Impairments. CVHI 2007.*