Bern University of Applied Sciences
Engineering and Information Technologies

TrustVote: A Proposal for a Hybrid E-Voting System

Dr. Rolf Haenni, Reto Koenig, Dr. Stephan Fischli, Dr. Eric Dubuis

Research report BFH-TI Nr. 6 of 1 August 2009

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

Research reports published previously

Nr. 5 Research on E-Voting Technologies. Dr. Rolf Haenni, Dr. Eric Dubuis, Dr. Ulrich Ultes-Nitsche. October 2008.

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

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

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

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

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

Concept and execution
Jens Geisel

Editorial office
Author(s) and Jens Geisel

Bern University of Applied Sciences
Engineering and Information Technologies

TrustVote: A Proposal for a Hybrid E-Voting System

Dr. Rolf Haenni, Reto Koenig, Dr. Stephan Fischli, Dr. Eric Dubuis

Research report BFH-TI Nr. 6 of 1 August 2009

Abstract

This paper presents a hybrid e-voting system, in which a transparent e-voting protocol is embedded in a traditional paper-based voting procedure. To guarantee vote anonymity, the protocol itself is based on a scalable blind signature scheme with multiple authorities. An anonymous channel is used to cast the encrypted votes onto the public board. To prevent vote buying and vote coercion, we depart from the mainstream approach of taking additional measures to guarantee receipt-freeness. Instead, we propose to exploit the existence of a receipt to allow vote revocations over the enclosing paper-based voting procedure.

Key words

remote e-voting, hybrid e-voting system, blind signatures, secret sharing, threshold signature scheme, security, privacy, cryptography

Biel/Bienne, August 2009

Authors

Prof. Dr. Rolf Haenni
Mobile Information Society: Identity, Security, Privacy.
Bern University of Applied Sciences, Engineering and Information Technology.
rolf.haenni (at) bfh.ch

Prof. Reto Koenig
Mobile Information Society: Identity, Security, Privacy.
Bern University of Applied Sciences, Engineering and Information Technology.
reto.koenig (at) bfh.ch

Prof. Dr. Stephan Fischli
Mobile Information Society: Identity, Security, Privacy.
Bern University of Applied Sciences, Engineering and Information Technology.
stephan.fischli (at) bfh.ch

Prof. Dr. Eric Dubuis
Mobile Information Society: Identity, Security, Privacy.
Bern University of Applied Sciences, Engineering and Information Technology.
eric.dubuis (at) bfh.ch

TrustVote: A Proposal for a Hybrid E-Voting System

Rolf Haenni, Reto Koenig, Stephan Fischli, and Eric Dubuis

Bern University of Applied Sciences, Höheweg 80, CH-2501 Biel
{rolf.haenni,reto.koenig,stephan.fischli,eric.dubuis}@bfh.ch

Abstract. This paper presents a hybrid e-voting system, in which a transparent e-voting protocol is embedded in a traditional paper-based voting procedure. To guarantee vote anonymity, the protocol itself is based on a scalable blind signature scheme with multiple authorities. An anonymous channel is used to cast the encrypted votes onto the public board. To prevent vote buying and vote coercion, we depart from the mainstream approach of taking additional measures to guarantee receipt-freeness. Instead, we propose to exploit the existence of a receipt to allow vote revocations over the enclosing paper-based voting procedure.

1 Introduction

Governments around the world are increasingly considering the replacement of traditional paper-based voting schemes with electronic voting systems. A particular form of such *e-voting* systems are those which allow voters to cast their votes over the internet, so-called *remote e-voting* or *i-voting* systems. In this paper, we will use the general term *e-voting* in a very restricted sense for remote e-voting over the internet, and with *voting* we refer to elections, referenda, polls, and other forms of collective decision making.

The idea of introducing electronic means into the electoral process has generated a lively debate, in which e-voting is viewed both a chance and a danger for democracy. The hope of e-voting enthusiasts includes the possibility of positive effects such as higher voter participation, improved pre-electoral opinion formation, or increased cost-effectiveness, whereas the fears of sceptics are mostly tied to security concerns and the resulting possibility of large-scaled frauds. The legitimacy of such security concerns has been demonstrated by the negative e-voting experience in the Netherlands, where all nationwide e-voting activities have been stopped in 2007 after the vulnerability of the deployed system had been exposed in public [1].

1.1 Requirements

For an e-voting system to be secure, it has to function without vulnerabilities in potentially insecure environments such as the internet. For this, it has to be implemented according to an intrinsically secure design. Despite the complexity of designing and implementing such a system, some criteria seem to be unanimously accepted as the *core security requirements* for e-voting systems [2, 3]:

Accuracy: A systems is accurate if casted votes can not be altered (*integrity*), valid votes can not be eliminated from the final tally (*completeness*), and invalid votes are not counted in the final tally (*soundness*).

Democracy: A system is democratic if only authorized voters can vote (*eligibility*) and eligible voters can only vote once (*uniqueness*).

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

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

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

The literature on e-voting technologies offers various *protocols* to establish these core requirements. Note that some requirements seem to be inherently contradictory, e.g. individual verifiability appears to be incompatible with receipt-freeness [4, 5]. The latter is important as a protection against vote buying and vote coercion [6].

Further requirements, which address general security properties of an implemented system, are less specific to e-voting but still crucial for introducing remote e-voting in practice. Examples of such general system requirements are *availability*, *reliability*, *resumability*, *robustness*, *accountability*, *auditability*, *disclosability*, *persistence*, or *transparency* [7]. Recent critical voices have particularly called for disclosability (e.g. in form of open-source software) and transparency (e.g. by publishing casted votes on a public board) as key issues for establishing general confidence in e-voting technologies [8–10]. In Germany, the lack of transparency and untraceability of the deployed voting machines has led the Federal Constitutional Court to declare their use as unconstitutional [11].

Apart from the above security and system requirements, there are further desirable properties such as *convenience*, *flexibility*, *scalability*, or *mobility* [2]. As they are directly influencing the efficiency, usability, and trustworthiness of an e-voting system, they are indirectly affecting the security of the voting. An important requirement of that kind is the *vote-and-go* property, which allows voters to cast their votes in one single phase [12]. Further requirements, which address political, administrative, or juridical questions, are also very important for introducing e-voting in practice, but discussing these issues is beyond the scope of this paper.[1]

1.2 Related Work

One of the central technical challenges of designing an e-voting protocol is to simultaneously authenticate voters unequivocally while preserving the anonymity of their votes. The design of such a protocol usually involves several strong cryptographic primitives. Besides the regular application of encryption to establish confidential channels and digital signatures to ensure the integrity and authenticity of the transmitted messages, there are at least three major design approaches for e-voting protocols: protocols

[1] For a detailed list of legal and operational standards as defined by the European Union, we refer to [13, 14].

based on *blind signatures* [15–17], *anonymous channels* [18, 19], and *homomorphic encryption* [4, 20–22]. More recent is a proposal based on *linkable ring signatures* [12].

Each of the above design paradigms has its own advantages and disadvantages with respect to achieving the aforementioned security and systems requirements. What most of the existing approaches have in common is a default distrust assumption towards all involved parties. Therefore, the goal is to protect individual votes not only against external attacks by hackers or malicious software, but also against any reasonably sized coalition of internal attackers. For this, many protocols propose the replication of potential single points of failures with corresponding threshold parameters.

Another common problem of the above design paradigms is the inherent difficulty of achieving universal verifiability *and* receipt-freeness. The first proposal for a receipt-free e-voting protocol in [4] has later been disproved [22]. Quite a few receipt-free protocols have been developed since then, but most of them are based on impractical physical assumptions, procedural constraints, or trusted third parties [22–29]. A promising approach is the one based on linkable ring signatures [12], but since casting a single vote requires the public keys of all potential voters, it is not suitable for large-scale elections. There is also some theoretical research towards a better understanding of receipt-freeness and coercion-resistance [6, 30].

Applying blind signatures to e-voting has first been proposed in [17]. In the suggested protocol, known as FOO92, the voter encrypts the vote first and then requests a blind signature from the voting authority. The blind signature ensures that the content of the vote remains entirely disguised from the voting authority during the authorization process. The encrypted vote together with the blind signature is then sent over an anonymous channel to a public board. To open the votes for counting, the voter supplies the encryption key at the end of the voting period, again over an anonymous channel. FOO92 has three major drawbacks. First, it contains potential single points of failure, e.g. it allows the authority to introduce votes for voters who abstain from casting their votes. Second, as voters need to be active in more than one phase of the protocol, it does not provide the desired vote-and-go property. Finally, as it does not offer receipt-freeness, it leaves the doors open for vote buying and vote coercion.

To overcome these drawbacks, many variations of the FOO92 protocol have been suggested in the literature [2, 23, 31–33], and several prototype implementations have been realized. One of the first and most-cited prototype systems is Sensus, which has been implemented and tested at the Washington University [34]. Similar implementations are Evox, which has been used for campus-wide elections at the MIT [35, 36], and Votopia, which has been built for the FIFA WorldCup 2002 in Korea/Japan to select the most valuable players [37]. More recent implementations of the same type of blind-signature based protocols are Revs [38], Seas [39], Civitas [40], and two others with no particular name [41, 42]. The development of these systems is stimulated by the fact that blind signature schemes are simple to understand and implement, flexible to be adjusted to all sorts of settings, and suitable for large-scale elections.

1.3 Contribution and Overview

This paper presents a hybrid e-voting system, called *TrustVote*, which consists of a paper-based and an e-voting component. The paper-based component realizes a traditional paper-based voting procedure, in which legitimate voters are supposed to physically drop their ballots into a ballot box. For this, we need to assume that a classical paper-based voting infrastructure is available and trustworthy. Our system is thus designed to be used by governments or organizations, which intend to enhance rather than to replace the existing voting infrastructure by an e-voting system.

An important feature of the proposed hybrid system is that it allows to revoke a previously casted electronic vote by subsequently casting an additional paper vote. This is similar to the "re-vote" feature of the Estonian e-voting system [43, 44], and it corresponds to the general idea of counting the "last ballot" only [6]. The revocation procedure is initiated by revealing the receipt obtained from the e-voting component to the voting authority conducting the paper-based procedure. Our approach thus departs from the mainstream approach of taking additional measures in the electronic voting protocol to guarantee receipt-freeness. In contrast, we propose to exploit the receipt as a vote revocation identifier to protect the hybrid system against vote buying and vote coercion.

The TrustVote e-voting component is an improved protocol based on blind signatures, similar to FOO92 and its successors. It defines the entities involved in the e-voting procedure, the communication between them, the exchanged data, and the processing of this data by each entity. With the (intended) exception of receipt-freeness, the bare protocol guarantees all core security requirements (accuracy, democracy, anonymity, individual and universal verifiability, fairness). It furthermore provides transparency and scalability, as well as the desired vote-and-go property. The robustness, availability, and reliability of the protocol is increased by distributing the responsibility to multiple registration authorities and key collectors. Both, the TrustVote protocol as well as the entire hybrid TrustVote system, are therefore suitable for real large-scale elections.

The remainder of this paper is organized as follows. We start with some cryptographic preliminaries and notational conventions in Section 2. Then the TrustVote e-voting protocol is discussed and analyzed in Section 3, and the hybrid TrustVote system in presented in Section 4. We summarize and conclude the paper in Section 5.

2 Cryptographic Preliminaries and Notation

The TrustVote e-voting protocol of the next section consists of multiple phases, in which different entities exchange e-voting data. Here, we describe the cryptographic functions used by these entities to process the exchanged data and different types of channels over which the data is transferred.

2.1 Basic Cryptographic Functions

In order to guarantee confidentiality and to provide authenticity, integrity, and non-repudiation, we use symmetric encryption and digital signatures, and denote the corresponding functions as follows:

$\mathrm{encrypt}_k(x)$	Encrypt data x using the secret key k;
$\mathrm{decrypt}_k(x)$	Decrypt data x using the secret key k;
$\mathrm{sign}_d(x)$	Sign data x using the private key d;
$\mathrm{verify}_e(s, x)$	Verify the signature s of x using the public key e.

Furthermore, we use a pseudo-random number generator to create secrets, an injective one-way function to create anonymous identifiers, and a cryptographic hash function to produce fixed-sized digests from arbitrary blocks of data:

random()	Generate a random number;
one_way(x)	Compute a unique one-way value of some input;
hash(x)	Compute a hash value of some data.

2.2 Blind Signature

A *blind signature*, as introduced by Chaum [15], is a form of digital signature, where the signer is not supposed to see what he or she is signing. In order to achieve this goal, the data to be signed is disguised before it is given to the signer using a blinding function. This function usually involves the public key e of the signer and a random number r:

$$x' = \mathrm{blind}_e(x, r).$$

After the signer has signed the blinded data x' using his or her private key d, the resulting blind signature s' can be transformed to an ordinary digital signature s using a corresponding unblinding function:

$$\begin{aligned} s' &= \mathrm{sign}_d(x'), \\ s &= \mathrm{unblind}(s', r). \end{aligned}$$

There are different blind signature schemes, one of the simplest is using classical RSA-based signatures. In this scheme, the blinding and unblinding functions consist of multiplying x with the blinding factor r^e and s' with the unblinding factor r^{-1}, respectivly.

2.3 Threshold Blind Signature

A (t, N)-*threshold signature* scheme is a procedure to let N parties sign some data such that the outcome is a valid signature if at least t parties have contributed to the signature [45]. We realize such a scheme by having each party sign the data individually and then counting the number of valid signatures in order to decide if the threshold has been achieved. If $\mathbf{s} = (s_1, \ldots, s_N)$ denotes the individual signatures and $\mathbf{e} = (e_1, \ldots, e_N)$ the public keys of the signers, we denote the corresponding verification function by

$$\mathrm{verify}_{\mathbf{e},t}(\mathbf{s}, x) \in \{\mathit{true}, \mathit{false}\}.$$

A *threshold blind signature* scheme is a combination of a threshold signature scheme with blind signatures such that the data to be signed is not revealed to the signers [46, 47]. Since it is important in our protocol that each party is given the same blinded data x', we assume that both the blinding and the unblinding function,

$$\begin{aligned} x' &= \mathrm{blind}_{\mathbf{e}}(x, r), \\ \mathbf{s} &= \mathrm{unblind}_{\mathbf{e}}(\mathbf{s}', r), \end{aligned}$$

depend on the public keys $\mathbf{e}$ of all the signing parties.

To realize such a scheme based on RSA, we can use a common blinding factor $r^{e_1 \cdots e_N}$ and individual unblinding factors $r^{-(e_1 \cdots e_{i-1} e_{i+1} \cdots e_N)}$ to obtain classical RSA signatures $s_i = x^{d_i}$. Note that if m_i denotes the modulus for the public key e_i, then $m_1 \cdots m_N$ will be an appropriate modulus for $r^{e_1 \cdots e_N}$.

2.4 Secret Sharing

A (t, N)-*threshold secret* sharing scheme is a procedure to distribute some secret data x among N parties. For this, each party is given a share x_i of the data such that any subgroup of t or more parties can reassemble the secret data, but no group of fewer than t parties can. We denote the sharing and assembling functions by

$$\mathbf{x} = \mathrm{share}_{t,N}(x),$$
$$x = \mathrm{assemble}_t(\mathbf{x}'),$$

respectively, where $\mathbf{x}'$ is a subset of the shares $\mathbf{x} = (x_1, \ldots, x_N)$ of size t or greater.

One of the first secret sharing schemes was introduced by Shamir [48]. It relies on the fact that a polynomial function $f(z) = a_0 + a_1 z + \cdots + a_{t-1} z^{t-1}$ of degree $t-1$ is determined by t points. Taking the secret data as the first coefficient a_0 and choosing the remaining coefficients $a_1, \ldots, a_{t-1}$ at random, any N points $(z, f(z))$ are distributed as shares amoung the N parties. Given any subset of the t points, the coefficients of f can be determined using interpolation, especially the secret data a_0.

2.5 Confidential, Authentic, and Anonymous Channels

In order to protect data during the transfer between the entities, we distinguish different types of communication channels. A *confidential channel* is a way of transferring data that is resistant to interception, whereas an *authentic channel* is resistant to tampering. Assuming the existence of a public key infrastructure, such channels can be established by using asymmetric encryption and digitial signatures, respectively.

An *anonymous channel* hides the correspondence between senders and their messages, i.e., the senders of the messages are anonymous or untraceable. The first anonymous channel, called *mix net*, was proposed by Chaum [18]. A mix net consists of a sequence of servers, each of which receives a batch of input messages and produces a batch of output messages in a permuted (mixed) order. In the formal description of the protocol, we will use the following notations for the transfer of data over the different types of channels:

$X \longrightarrow Y\colon\ x$ Transfer x over an unsecure channel;
$X \Longrightarrow Y\colon\ x$ Transfer x over a confidential channel;
$X \mathrel{\|\!\!\longrightarrow} Y\colon\ x$ Transfer x over an authentic channel;
$X \mathrel{-\!\!\!|\!|\!|\!\!\!\rightarrow} Y\colon\ x$ Transfer x over an anonymous channel;
$X \mathrel{\|\!\!\Longrightarrow} Y\colon\ x$ Transfer x over a confidential and authentic channel;
$X \mathrel{=\!\!\!|\!|\!|\!\!\!\Rightarrow} Y\colon\ x$ Transfer x over a confidential and anonymous channel.

For the transfer of data to multiple recipients, we write $X \longrightarrow \mathbf{Y}:\ x$ if the data is the same for every recipient and $X \longrightarrow \mathbf{Y}\colon\ \mathbf{x}$ if the data is different. Similarly, $\mathbf{X} \longrightarrow Y\colon\ \mathbf{x}$ denotes the transfer of data from multiple senders to a single recipient.

3 The TrustVote E-Voting Protocol

The TrustVote e-voting protocol as described in this section is an enhancement of the original FOO92 protocol [17]. The proposed extensions are necessary to eliminate the problems mentioned at the end of Subsection 1.2. Some of the extensions are adopted from the successors of FOO92, most notably from Sensus [34], Evox [35], and Revs [38]. An additional extension addresses the possibility of injecting additional votes by a conspiring group of voters, a possible vulnerability that has been overseen in Revs.[2]

One of the main difference between FOO92 and TrustVote is the transition from a single blind signature to a threshold blind signature scheme. For this, the administration's task is reduced to the initialization of the voting process, whereas the responsibility for the voter registration process is transferred to a group of so-called registration authorities, each of them with the ability to issue blind signatures. To render a casted vote valid, it must then be equipped with a reasonable number (defined by a threshold) of such signatures. Such a multiple authority scheme has first been proposed as an Evox enhancement [36] and later in Revs. TrustVote differs from those approaches by interlinking the individual blind signatures with a common blinding factor (see Subsection 2.3). To ensure the uniqueness of the common blinding factor, the protocol requires an additional public board. This enhancement is important to avoid the above-mentioned vulnerability with respect to a conspiring group of voters and to allow arbitrary thresholds. Note that we use the same additional public board to embed the TrustVote protocol into the hybrid system (see Section 4).

The second important difference between FOO92 and TrustVote concerns the opening of the encrypted votes. For this, FOO92 requires an additional voter interaction and thus rules out the desired vote-and-go property. Sensus and Evox suggest to release the decryption keys together with the casted votes and to delegate the responsibility for holding back the keys during the voting period to the "tallier" or the "anonymous server". This is similar to Revs, where all the votes are encrypted with the public key of the "commissioner", which is supposed to unveil the corresponding private key at the end of the voting period. Note that the fairness property is violated in either case. To avoid this type of problem, TrustVote uses a secret sharing algorithm to protect the secrecy of the decryption keys during the vote casting period (see Section 2.4).

3.1 Roles and PKI Setup

In the following, we introduce the entities involved in the protocol and their roles. Further details about their roles will be explained in the subsequent protocol description.

Voter: V

The voter is the main actor of the protocol. In a real-world setting, there will be a large number of voters, but the subsequent description of the protocol only involves the perspective of a single one. With *id* we denote his or her unique identifier.

Administration: D

The administration prepares and initiates the voting process. This involves compiling and publishing the list of legitimate voters, creating the official electronic ballot, and specifying a unique event number.

[2] Many thanks to our colleague Emmanuel Benoist for pointing this out.

Registration Authorities: $\mathbf{A} = (A_1, \ldots, A_N)$

The registration authorities are responsible for verifying the eligibility of a potential voter to vote and for preventing them to vote more than once. Since this is one of the most crucial tasks of the whole protocol, we propose to share this responsibility among $N \geq 2$ different registration authorities. The idea then is to choose N according to the number of opposing parties or candidates involved in the election, and to let each one of them control or supervise one registration authority.

Key Collectors: $\mathbf{C} = (C_1, \ldots, C_M)$

The key collectors are responsible for the non-disclosure of the vote decryption keys during the vote casting phase. To avoid a potential single point of failure, we assume again that there are at least $M \geq 2$ key collectors and that M is chosen according to the number of involved parties or candidates.

Public Boards: R, B

A public board is a broadcast channel with memory. This means that all entities are allowed to append new entries and to read its content, but nobody is allowed to delete or to modify existing entries. Such a board may have the additional functionality of filtering out invalid or double entries, e.g. by checking the validity of an attached digital signature. Our protocol involves two public boards with different responsibilities: the *registration board R* for listing the voters during the registration process and the *voting board B* for publishing the casted votes. The boards can be replicated in order to prevent them from being potential single points of failure, but we do not explicitly include this replication as part of the protocol.

Tallier: T

The tallier counts the published votes at the end of the voting process. Note that this is no explicit entity of the protocol, as any entity can take on the role of a tallier at the end of the voting period.

In the subsequent protocol description, we assume the existence of a public key infrastructure (PKI), which includes the administration, the authorities, the key collectors, and all legitimate voters (but not the public boards). The corresponding public keys shall publicly be available in form of certificates issued by a trustworthy certification authority. We denote the private and public keys of the authorities $\mathbf{A}$ by $\mathbf{d} = (d_1, \ldots, d_N)$ and $\mathbf{e} = (e_1, \ldots, e_N)$, respectively, and the key pair of a voter V by (d, e).

Assuming a complete PKI simplifies the registration phase of the protocol, but it also restricts the applicability of the protocol in practice. Note that the protocol can easily be turned into a non-PKI version, in which voters obtain a temporary key pair after identifying themselves with their PIN or password.

3.2 The Protocol

The TrustVote e-voting protocol consists of five consecutive phases, which will now be explained in detail. The description of the protocol only involves the perspective of a single voter V. A first overview of the whole protocol is given in the sequence diagram shown in Figure 1. Note that the voter preparation phase is invisible in the diagram, since it does not involve any communication.

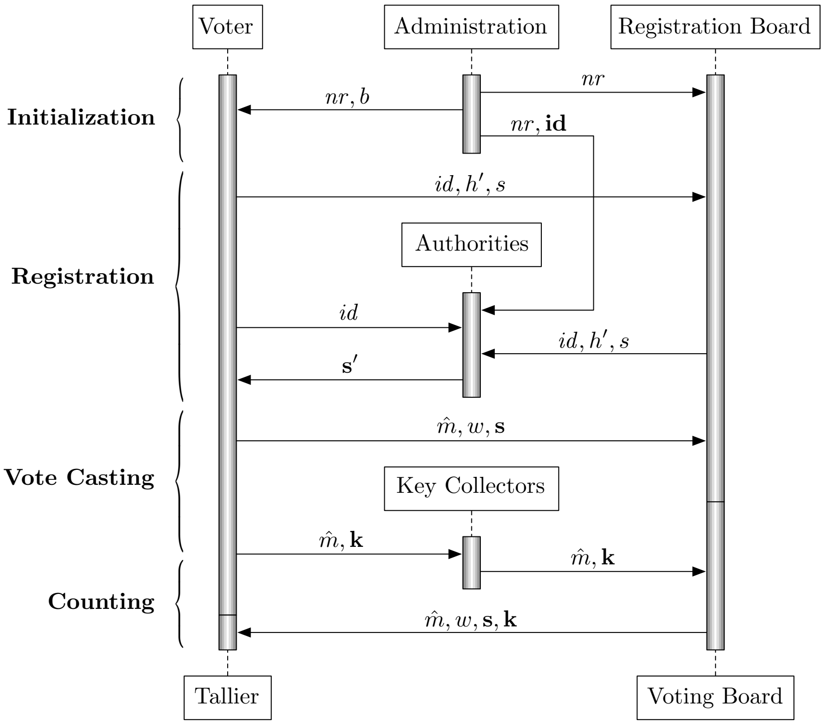

Fig. 1. The four communication phases of the TrustVote e-voting protocol. The voter preparation phase, which takes place between the initialization and the registration phase, is not visible.

Phase 1: Initialization The administration D initiates the voting process by generating a unique event number nr and the empty ballot b. Both, the event number and the ballot, are distributed over an authentic channel to all necessary entities, e.g. by publishing them together with respective digital signatures on an official web site. Furthermore, the administration sends the list of identifiers of legitimate voters (denoted by **id**) over an authentic channel to the voting authorities **A**. At the end of the preparation phase, all general parameters are specified and publicly known. The role of the administration as an active protocol entity terminates.

1.1 $D \Vdash\!\!\longrightarrow V, \mathbf{A}, R, B, T:\ nr$
1.2 $D \Vdash\!\!\longrightarrow V, T:\ b$
1.3 $D \Vdash\!\!\longrightarrow \mathbf{A}:\ \mathbf{id}$

Phase 1: Initialization

Phase 2: Voter Preparation At the beginning of the second phase, the voter fills in the empty ballot b and the resulting vote v is encrypted with a randomly chosen secret key k. Then the so-called *voter mark* $\hat{m}$ is generated by applying a one-way function to $id \,\|\, nr \,\|\, m$, where m is the randomly chosen *voter secret*. The role of $\hat{m}$ is the one of a self-assigned pseudonym, which is unique for the current and all future voting events. Note that no communication takes place during this phase, i.e., everything happens locally on the voter's machine.

At the end of this phase, the voter is ready to start the registration and vote casting processes, in which the voter mark $\hat{m}$ will serve as an anonymous identifier on the voting board. During the vote revocation process in the embedding hybrid system (see Section 4.1), we will use the self-assigned voter secret m as a receipt for proving the link between the voter (as registered on the registration board) and the vote (as casted on the voting board).

```
2.1  $V:\ v = \text{fill\_ballot}_V(b)$
2.2  $V:\ k = \text{random}()$
2.3  $V:\ w = \text{encrypt}_k(v)$
2.4  $V:\ m = \text{random}()$
2.5  $V:\ \hat{m} = \text{one\_way}(id \,\|\, nr \,\|\, m)$
```

Phase 2: Voter Preparation

Phase 3: Registration The purpose of the registration phase is to authorize legitimate voters to cast their votes. For this, the voter requests from the authorities a blind signature. To initiate the blind signature scheme, the voter calculates a verification code $h = \text{hash}(\hat{m} \,\|\, nr \,\|\, w)$ to interlink the encrypted vote w with the voter mark and the current event number. The voter then blinds the verification code using the blinding function described in Subsection 2.3. The resulting value h' is the common data to be signed by the authorities. To prevent the voter from sending different values to the authorities, we use the registration board as a broadcast channel for h'. To avoid false or multiple entries on the board, only those with a valid voter signature are accepted (one entry per voter). The authorities can then respond to a voter's request for a blind signature by retrieving h' from the board and signing it with their own private keys. The request itself is initiated by sending the voter's identifier *id* over an authentic channel to the authorities.

The registration phase ends as soon as the number of returned blind signatures exceeds a certain threshold t. To avoid a single point of failure, t must be greater than 1 and smaller than N. To maximize the robustness and reliability of the protocol, the choice of t should make it unlikely that t or more authorities collude, or that $N - t$ authorities fail. For situations like this, $\frac{2}{3}N \leq t \leq \frac{3}{4}N$ is often mentioned as a reasonable choice.

```
$V\colon\ r = \mathrm{random}()$
$V\colon\ h = \mathrm{hash}(\hat{m} \| nr \| w)$
$V\colon\ h' = \mathrm{blind}_{\mathbf{e}}(h, r)$
$V\colon\ s = \mathrm{sign}_d(id \| nr \| h')$
$V \longrightarrow R\colon\ id, h', s$
$\textbf{if } id \notin R \textbf{ and } \mathrm{verify}_e(s, id \| nr \| h') = \mathit{true} \textbf{ then}$
    $R\colon\ \mathrm{publish}(id, h', s)$
$V \Vdash\!\!\!\longrightarrow \mathbf{A}\colon\ id$
$\textbf{if } id \in \mathbf{id} \textbf{ then}$
    $R \longrightarrow \mathbf{A}\colon\ id, h', s$
    $\textbf{if } \mathrm{verify}_e(s, id \| nr \| h') = \mathit{true} \textbf{ then}$
        $\mathbf{A}\colon\ \mathbf{s}' = \mathrm{sign}_{\mathbf{s}}(h')$
        $\mathbf{A} \Vdash\!\!\!\longrightarrow V\colon\ \mathbf{s}'$
        $V\colon\ \mathbf{s} = \mathrm{unblind}_{\mathbf{e}}(\mathbf{s}', r) = \mathrm{sign}_{\mathbf{d}}(h)$
```

Phase 3: Registration

Phase 4: Vote Casting In this phase, the voter first applies the secret sharing algorithm $\mathrm{share}_{u,M}(k)$ to distribute the decryption key k as a set $\mathbf{k} = (k_1, \ldots, k_M)$ of corresponding shares (see Subsection 2.4), where u denotes the minimal number of shares needed to reassemble the key. Choosing the threshold u with care is again important to maximize the robustness and reliability of the protocol. The shares are then sent anonymously to the key collectors **C** (one share per collector).

Then the encrypted vote together with the authorities' signatures **s** are deposited anonymously on the voting board B, the voter mark serving as an anonymous identifier. To avoid false entries on the board, the integrity of the received data is checked by calculating the verification code $h = \mathrm{hash}(\hat{m} \| nr \| w)$ and by verifying the validity of the supplied signatures. Only entries with at least t valid signatures are accepted. The voter can now verify that his or her vote has successfully been casted.

At the end of this phase, the voting board has published the encrypted vote together with the signatures, and the key collectors are in possession of the shared secret key. With this, the role of the voter as active protocol entity terminates.

```
$V\colon\ \mathbf{k} = \mathrm{share}_{u,M}(k)$
$V \mathrel{\#\!\!\Rightarrow} \mathbf{C}\colon\ \hat{m}, \mathbf{k}$
$V \mathrel{\#\!\!\rightarrow} B\colon\ \hat{m}, w, \mathbf{s}$
$B\colon\ h = \mathrm{hash}(\hat{m} \| nr \| w)$
$\textbf{if } \hat{m} \notin B \textbf{ and } \mathrm{verify}_{\mathbf{e},t}(\mathbf{s}, h) = \mathit{true} \textbf{ then}$
    $B\colon\ \mathrm{publish}(\hat{m}, w, \mathbf{s})$
```

Phase 4: Vote Casting

Phase 5: Counting The last phase of the TrustVote e-voting protocol involves the opening of the encrypted votes to make them available for counting. For this, the collectors send their shares of the secret key over an authentic channel to the voting board, which completes the existing entries using the voter marks as identifiers. The votes are now ready to be counted.

As anyone can read the content of the voting board, we do not further specify the entity that plays the role of the tallier T. To prepare the counting, each vote with enough valid signatures is decrypted with the reassembled decryption key $k = \text{assemble}_u(\mathbf{k})$. After successfully comparing the format and content of a decrypted vote with the original empty ballot, it is accepted as a valid vote. The result of the counting will be used as such in the hybrid system (see Subsection 4.1).

```
$\mathbf{C} \Vdash\!\!\longrightarrow B$: $\hat{m}, \mathbf{k}$
if $(\hat{m}, w, \mathbf{s}) \in B$ then
    $B$: publish$(\hat{m}, w, \mathbf{s}, \mathbf{k})$
$B \longrightarrow T$: $\hat{m}, w, \mathbf{s}, \mathbf{k}$
$T$: $h = \text{hash}(\hat{m} \| nr \| w)$
if $\text{verify}_{\mathbf{e},t}(\mathbf{s}, h) = \mathit{true}$ then
    $T$: $k = \text{assemble}_u(\mathbf{k})$
    $T$: $v = \text{decrypt}_k(w)$
    if $v$ is consistent with $b$ then
        $T$: count$(v)$
```

Phase 5: Counting

3.3 Security Analysis

Let us now have a closer look at the security properties of the TrustVote protocol. For this, we assume that a secure implementation of the cryptographic functions and channels described in Section 2 exists, and that the machines used by the voters are invulnerable in the sense of a *secure platform* [49]. We also assume that none of the involved private keys has been lost or stolen. Furthermore, we suppose that at least t registration authorities and u key collectors are reliable and willing to co-operate, and that the list of legitimate voters distributed by the administration is sound and complete. Then the TrustVote protocol fulfills all the core security requirements except for receipt-freeness:

Integrity: Since all casted votes are published on the voting board, any alteration would immediately be detected. Such an alteration could easliy be proved, because the authorities' signatures would become invalid.

Completeness The observability of the voting board also guarantees that no votes can be removed without being noticed.

Soundness: Unauthorized votes, i.e., votes with an insufficient number of valid signatures, are rejected by both the voting board and the tallier. Invalid votes, which are inconsistent with the ballot, will be recognized during the counting phase and thus will be ignored for the final tally.

Eligibility: The registration authorities only accept signing requests from voters, which are on the list of legitimate voters distributed by the administration. Therefore, only legitimate voters will be able to cast their votes on the voting board.

Uniqueness: To obtain the blind signatures from the authorities, the voter has to publish the blinded verification code (which depends on the voter mark and the encrypted vote) on the registration board. Since the registration board only accepts one entry per voter, no voter can obtain signatures for more than one vote.

Anonymity: The blind signature scheme prevents the authorities from learning the voter's verification code, which implies that they cannot associate the voter with a voter mark or a vote. After the registration phase, the voter uses the self-assigned voter mark as a pseudonym to anonymously publish the vote on the voting board and to send the shares of the decryption key to the key collectors. The anonymous channels guarantee that both, the voting board and the key collectors, cannot trace back the cast to the voter.

Individual Verifiability: The voter mark identifies the casted vote on the voting board and can thus be used by the voter to independently verify that his or her own vote is included in the final tally.

Universal Verifiability: The public boards allow any entity to independently verify that all casted votes are counted correctly.

Fairness: Since the casted votes remain encrypted during the voting period, no intermediate result can be inferred from to information available before the key collectors reveal the decryption keys at the end of the voting period.

Since the voter mark serves as an identifier for a casted vote, it is always possible to create an undeniable link to a particular vote by revealing the voter secret. This implies that the TrustVote protocol is obviously not receipt-free. But we will see in Section 4.2 how to exploit the receipt in the vote revocation process to overcome vote buying and vote coercion problems in the hybrid system.

Apart from the above core security requirements, the TrustVote protocol also fulfills some other important requirements. The redundancy established by the threshold signature and secret sharing schemes makes the system robust against groups of colluding or failing entities up to a certain size (defined by the thresholds t and u). Since the encrypted votes and the corresponding shares of the decryption keys are casted in the same phase of the protocol, the voter needs to interact with the system only once and can thus cast the vote in a vote-and-go manner. The protocol also offers some additional measures to prevent denial-of-service attacks (e.g. by flooding of the public boards), which improves the overall availability and reliability of the system. Finally, the protocol offers complete transparency by publishing the casted votes on public boards.

4 The Hybrid E-Voting System

In this section, we show how to embed the e-voting protocol of the previous section into a traditional paper-based voting infrastructure. The goal is to offer the voters the possibility to revoke previously casted electronic votes by presenting their voter secrets to the voting officials. If a revealed voter secret matches with a voter mark on the voting board, the corresponding electronic vote counts as a negative vote and is subtracted from the final result. The voter is then allowed to cast the vote update on paper, the so-called "last ballot" [6]. An overview of the hybrid voting process is shown in the data flow diagram of Figure 2.

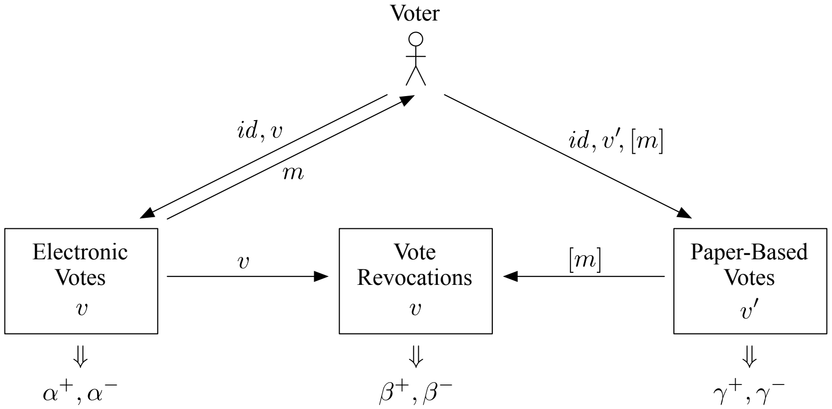

Fig. 2. Data flow diagram of the hybrid e-voting systems. The optional revealing of the voter secret *m*, denoted by [*m*], initiates the vote revocation process.

The idea of merging an electronic voting system with a traditional paper-based voting infrastructure is not entirely new in the literature. Some papers advocate a so-called *voter-verifiable paper trail* (VVPAT) as backup to allow recounting in case of a fraud [50–53]. Another type of papers proposes so-called *end-to-end auditable voting systems*, in which paper ballots containing cryptographic elements are processed by optical scanners. Examples of such systems are *Prêt à Voter* [54], *Punchscan* [55], *Scantegrity* [56], *ThreeBallot* [57], or *Bingo Voting* [58]. They differ strongly from the type of hybrid system proposed in this paper, in which a traditional paper-based system coexists with its electronic counterpart. The potential of using a paper-based and an electronic system together, so that each can do what it does best, and each can compensate for the drawbacks of the other, has been mentioned in [6, §5.2], but it seems that this is still a relatively unexplored view in the literature.

4.1 Vote Revocations and Counting in the Hybrid System

We now describe how the vote revocation mechanism and the counting procedure works in the hybrid system. First, we assume that the electronic vote casting period ends *before* the paper-based procedure starts. This chronological constraint is important to ensure that revoking an electronic vote is always possible and that no electronic vote is accepted after casting a paper vote. In a real-world setting, electronic votes could be accepted a few days or a week prior to the official election day. The electronic voting process works as explained in the previous subsection, i.e., the encrypted votes are published on the voting board and the voters memorize their voter secrets. But to avoid the disclosure of intermediate results, the counting phase is postponed until all voting offices are closed and the paper-based voting period ends. Attaining this second chronological constraint is the responsibility of the key collectors.

Now suppose that a voter visits the voting office on the election day to cast a paper vote. In a traditional paper-based setting, the election officials ask the voter to prove his identity before allowing him to drop the ballot into the ballot box. In our hybrid system, some additional steps are needed at this point. The first one is to check if the voter has previously casted an electronic vote. For this, the officials check if the registration board R contains an entry with the voter's *id*. If no such entry exists, they allow the voter to cast the paper vote v' as usual.

If an entry with the voter's *id* exists on the registration board, the vote revocation process starts. For this, the voter is asked to reveal the voter secret m to the officials, who can then compute the corresponding voter mark $\hat{m} = \text{one\_way}(id \,\|\, nr \,\|\, m)$. If an entry for $\hat{m}$ is found on the voting board, then the vote revocation is accepted and the voter is allowed to cast the new paper vote v'. Otherwise, the vote revocation and thus the vote update are rejected. The voter secrets of all accepted vote revocations are collected on a separate pile.

At the end of the voting period, the hybrid vote counting phase starts. Without loss of generality, we assume here a simple *yes*/*no*-type of voting, i.e., each vote is one bit with two possible values *yes* and *no*. The problem then is to determine three pairs of values (see bottom of Figure 2):

$$\begin{aligned}
\alpha^+, \alpha^- &: \text{number of electronic votes for } \textit{yes} \text{ and } \textit{no}\text{, respectively;}\\
\beta^+, \beta^- &: \text{number of revoked votes for } \textit{yes} \text{ and } \textit{no}\text{, respectively;}\\
\gamma^+, \gamma^- &: \text{number of paper votes for } \textit{yes} \text{ and } \textit{no}\text{, respectively.}
\end{aligned}$$

The two values α^+ and α^- for the electronic votes are derived from the voting board and are thus universally verifiable (as explained in Section 3). The values β^+ and β^- are determined by the voting officials from the pile of revealed voter secrets and the corresponding votes published on the voting board. Finally, γ^+ and γ^- are the results obtained from counting the paper votes in the traditional way. The final results are then $result^+ = \alpha^+ - \beta^+ + \gamma^+$ for the total number of *yes*-votes and $result^- = \alpha^- - \beta^- + \gamma^-$ for the total number of *no*-votes. Note that the counting procedure as presented here is easily extendable to more general voting schemes with multiple candidates or multiple seats.

4.2 Security Analysis

The security of the electronic component of the TrustVote system has been discussed in Subsection 3.3. In principle, all the desired security properties are inherited into the hybrid system, but revealing the voter secret m to start the vote revocation process could possibly create a leak. In fact, revealing m together with the voter *id* to the officials on the election day means to give up the anonymity of the electronic vote. But since this affects exactly those voters which are allowed to cast a vote update on paper, it only compromises votes that will not be counted in the final tally. In a simple *yes*/*no*-type of voting, one could argue that revoking a *yes*-vote implies that the update will be a *no*-vote, and vice versa. But this conclusion remains speculative and does therefore not violate the anonymity of the vote update. Furthermore, by offering the option of casting the “last ballot” on paper, we provide a simple mechanism to compensate for the protocol’s missing receipt-freeness property and to overcome the resulting vote buying and vote coercion problems. To identify the vote on the voting board, a potential vote buyer may ask the voter to reveal the voter secret m, but there is no general and scalable way of verifying that the vote has not been revoked afterwards.

To ensure the other desired security properties with respect to the aforementioned vote revocation process, we do not require assumptions different from those of a traditional paper-based voting infrastructure. As we propose the e-voting protocol to be embedded in an existing voting infrastructure, we can simply enhance the corresponding operational and procedural standards to the vote revocation process. This includes the strict separation between the voter’s identity and the vote update on paper, e.g. by dropping the latter secretly into a ballot box after verifying the former. It is also possible to let official election observers supervise the vote revocation process or to conduct a recount in case of a suspected fraud.

5 Conclusion

As governments today are rather considering a gradual transition instead of an overnight switch from an all-paper to a all-electronic system, there may be an increasing demand for well-designed hybrid systems in which a traditional paper-based system coexists with its electronic counterpart. The approach proposed in this paper is such a hybrid system, in which a traditional paper-based voting infrastructure is enhanced with an e-voting protocol. The protocol itself is based on blind signatures and a secret sharing scheme. Potential single points of failures are avoided by replicating some of the entities involved in the protocol. We showed that the protocol guarantees all major security requirements, except that it delivers a receipt and thus allows vote buying and vote coercion. To render those receipts useless, we demonstrated how to embed the e-voting protocol into a traditional paper-based voting infrastructure.

References

1. Loeber, L.: E-voting in the Netherlands: from general acceptance to general doubt in two years. In Krimmer, R., Grimm, R., eds.: 3nd International Workshop on Electronic Voting. Lecture Notes in Informatics, Bregenz, Austria, Gesellschaft für Informatik E.V. (2008) 21–30
2. 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)
3. 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
4. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections. In: STOC’94, 26th Annual ACM Symposium on Theory of Computing, Montréal, Canada (1994) 544–553
5. Weber, S.: A coercion-resistant cryptographic voting protocol: Evaluation and prototype implementation. Diploma thesis, Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany (2006)

6. Skripsky, J.: Minimal models for receipt-free voting. Semester project, ETH Zürich (2002)
7. Neumann, P.G.: Security criteria for electronic voting. In: NCSC'93, 16th National Computer Security Conference, Baltimore, USA (1993) 478–482
8. Ataç, D., Kayapinar, B., Riedel, W.: e-Voting mit Open Source. Gutachten, Informatik und Gesellschaft, Technische Universität Berlin (2004)
9. Benoist, E., Anrig, B., Jaquet-Chiffelle, D.O.: Internet-voting: Opportunity or threat for democracy? In Alkassar, A., Volkamer, M., eds.: VOTE-ID'07, 1st International Conference on E-Voting and Identity. LNCS 4896, Bochum, Germany (2007) 29–37
10. Sietmann, R.: Transparenz, vertrauen und kontrolle. c't (20) (2008) 46–49
11. Ziegler, P.M.: Karlsruhe zieht Black-Box-Voting den Stecker. heise online **10** (2009)
12. Chow, S.S.M., Liu, J.K., Wong, D.S.: Robust receipt-free election system with ballot secrecy and verifiability. In: NDSS'08, 15th Network and Distributed System Security Symposium, San Diego, USA (2008) 81–94
13. Council of Europe: Legal, Operational and Technical Standards for e-Voting. Rec(2004)11. Council of Europe Publishing (2004)
14. Volkamer, M., McGaley, M.: Requirements and evaluation procedures for evoting. In: ARES'07, 2nd International Conference on Availability, Reliability and Security, Vienna, Austria (2007) 895–902
15. Chaum, D.: Blind signatures for untraceable payments. In: CRYPTO'82, 2nd International Cryptology Conference, Santa Barbara, USA (1982) 199–203
16. Chaum, D.: Blind signature system. In: CRYPTO'83, 3rd International Cryptology Conference, Santa Barbara, USA (1983) 153–156
17. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In Seberry, J., Zheng, Y., eds.: ASIACRYPT'92, Workshop on the Theory and Application of Cryptographic Techniques. LNCS 718, Gold Coast, Australia (1992) 244–251
18. Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms. Communications of the ACM **24**(2) (1981) 84–88
19. Park, C., Itoh, K., Kurosawa, K.: Efficient anonymous channel and all/nothing election scheme. In Helleseth, T., ed.: EUROCRYPT'93, Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology. LNCS 765, Lofthus, Norway (1993) 248–259
20. Benaloh, J.D.C.: Verifiable Secret-Ballot Elections. PhD thesis, Yale University, New Haven, USA (1987)
21. Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. European Transactions on Telecommunications **8**(5) (1997) 481–490
22. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In Goos, G., Hartmanis, J., van Leeuwen, J., eds.: EUROCRYPT'00, International Conference on the Theory and Applications of Cryptographic Techniques. LNCS 1807, Bruges, Belgium (2000) 539–556
23. Okamoto, T.: Receipt-free electronic voting schemes for large scale elections. In Christianson, B., Crispo, B., Lomas, T.M.A., Roe, M., eds.: 5th International Security Protocols Workshop. LNCS 1361, Paris, France (1997) 25–35
24. Magkos, E., Burmester, M., Chrissikopoulos, V.: Receipt-freeness in large-scale elections without untappable channels. In Schmid, B., Stanoevska-Slabeva, K., Tschammer, V., eds.: I3E'01, 1st IFIP Conference on towards the E-Society. Volume 202. (2001) 683–694
25. Lee, B., Boyd, C., Dawson, E., Kim, K., Yang, J., Yoo, S.: Providing receipt-freeness in mixnet-based voting protocols. In Goos, G., Hartmanis, J., van Leeuwen, J., eds.: ICISC'03, 6th International Conference on Information Security and Cryptology. LNCS 2971, Seoul, Korea (2003) 245–258
26. Acquisti, A.: Receipt-free homomorphic elections and write-in ballots. Technical Report 105, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA (2004)

27. Moran, T., Naor, M.: Receipt-free universally-verifiable voting with everlasting privacy. In Dwork, C., ed.: CRYPTO'06, 26th Annual International Cryptology Conference on Advances in Cryptology. LNCS 4117, Santa Barbara, USA (2006) 373–392
28. Xia, Z., Schneider, S.: A new receipt-free e-voting scheme based on blind signature. In: WOTE'06, IAVoSS Workshop on Trustworthy Elections, Cambridge, U.K. (2006) 127–135
29. Meng, B.: An internet voting protocol with receipt-free and coercion-resistant. In Miyazaki, T., ed.: CIT'07, 7th IEEE International Conference on Computer and Information Technology, Aizu-Wakamatsu City, Japan (2007) 721–726
30. Jonker, H.L., Vink, E.P.: Formalizing receipt-freeness. In: ISC'06, 9th Information Security Conference. LNCS 4176, Samos, Greece (2006) 476–488
31. Baraani-Dastjerdi, A., Pieprzyk, J., Safavi-Naini, R.: A practical electronic voting protocol using threshold schemes. Technical report, University of Wollongong, Department of Computer Science, Wollongong, Australia (1994)
32. Ohkubo, M., Miura, F., Abe, M., Fujioka, A., Okamoto, T.: An improvement on a practical secret voting scheme. In Mambo, M., Zheng, Y., eds.: ISW'99, 2nd International Workshop on Information Security. LNCS 1729, Kuala Lumpur, Malaysia (1999) 225–234
33. Ray, I., Ray, I., Narasimhamurthi, N.: An anonymous electronic voting protocol for voting over the internet. In: WECWIS'01, 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, San Jose, USA (2001) 188–191
34. Cranor, L.F., Cytron, R.K.: Sensus: A security-conscious electronic polling system for the internet. In: HICSS-30, 30th Hawaii International Conference on System Sciences. Volume 03., Maui, USA (1997) 561–570
35. Herschberg, M.A.: Secure electronic voting using the world wide web. Master's thesis, Massachusetts Institute of Technology, Boston, USA (1997)
36. DuRette, B.W.: Multiple administrators for electronic voting. Bachelor thesis, Massachusetts Institute of Technology, Boston, USA (1999)
37. Kim, K.: Killer application of PKI to internet voting. In: IWAP'02, 2nd International Workshop for Asia Public Key Infrastructures, Taipei, Taiwan (2002)
38. Joaquim, R., Zuquete, A., Ferreira, P.: REVS – a robust electronic voting system. In: IADIS International Conference e-Society 2003, Lisbon, Portugal (2003) 95–103
39. Baiardi, F., Falleni, A., Granchi, R., Martinelli, F., Petrocchi, M., Vaccarelli, A.: SEAS, a secure e-voting protocol: Design and implementation. Computers & Security **24**(8) (2005) 642–652
40. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: SP'08, 29th IEEE Symposium on Security and Privacy, Oakland, USA (2008) 354–368
41. Anane, R., Freeland, R., Theodoropoulos, G.: E-voting requirements and implementation. In: CEC'07, 9th IEEE Conference on E-Commerce Technology, Tokyo, Japan (2007) 382–392
42. Aeby, A., Wiget, M.: On-Line Meinungsumfragen. Diploma thesis, Bern University of Applied Sciences, Biel, Switzerland (2007)
43. Estonian National Electoral Committee: E-voting system overview. (2005)
44. Volkamer, M., Grimm, R.: Multiple casts in online voting: Analyzing chances. In Krimmer, R., ed.: 2nd International Workshop on Electronic Voting. Number P-86 in Lecture Notes in Informatics, Bregenz, Austria, Gesellschaft für Informatik E.V. (2006) 97–106
45. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme. In Desmedt, Y., ed.: PKC'03, 6th International Workshop on Theory and Practice in Public Key Cryptography. LNCS 2567, Miami, USA (2003) 31–46
46. Kim, J., Kim, K., Lee, C.: An efficient and provably secure threshold blind signature. In Kim, K., ed.: ICISC'01, 4th International Conference on Information Security and Cryptology. LNCS 2288, Seoul, South Korea (2002) 318–327

47. Cao, Z., Zhu, H., Lu, R.: Provably secure robust threshold partial blind signature. Science in China Series F: Information Sciences **49**(5) (2006) 604–615
48. Shamir, A.: How to share a secret. Communications of the ACM **22**(11) (1979) 612–613
49. Gerck, E., Neff, C.A., Rivest, R.L., Rubin, A.D., Yung, M.: The business of electronic voting. In Syverson, P.F., ed.: FC'01, 5th International Conference on Financial Cryptography. LNCS 2339, Grand Cayman, British West Indies (2001) 243—268
50. Mercuri, R.: A better ballot box? IEEE Spectrum **39**(10) (2002) 46–50
51. Yasinsac, A., Bishop, M.: The dynamics of counting and recounting votes. IEEE Security & Privacy **6**(3) (2008) 22–29
52. Gold, V.: Computer security expert reinforces acm recommendations for secure, reliable e-voting. ACM/Press Release (2007)
53. Ansari, N., Sakarindr, P., Haghani, E., Zhang, C., Jain, A.K., Shi, Y.Q.: Evaluating electronic voting systems equipped with voter-verified paper records. IEEE Security & Privacy **6**(3) (2008) 30–39
54. Chaum, D., Ryan, P.Y.A., Schneider, S.A.: A practical voter-verifiable election scheme. In de Capitani di Vimercati, S., Syverson, P.F., Gollmann, D., eds.: ESORICS'05, 10th European Symposium on Research in Computer Security. LNCS 3679, Milan, Italy (2005) 118–139
55. Fisher, K., Carback, R.T., Sherman, A.T.: Punchscan: Introduction and system definition of a high-integrity election system. In: WOTE'06, IAVoSS Workshop On Trustworthy Elections, Cambridge, U.K. (2006)
56. Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., Vora, P.: Scantegrity: End-to-end voter-verifiable optical- scan voting. IEEE Security & Privacy **6**(3) (2008) 40–46
57. Rivest, R.L., Smith, W.D.: Three voting protocols: ThreeBallot, VAV, and Twin. In: EVT'07, USENIX/ACCURATE Electronic Voting Technology Workshop, Boston, USA (2007)
58. Bohli, J.M., Müller-Quade, J., Röhrich, S.: Bingo voting: Secure and coercion-free voting using a trusted random number generator. In Alkassar, A., Volkamer, M., eds.: VOTE-ID'07, 1st International Conference on E-Voting and Identity. LNCS 4896, Bochum, Germany (2007) 111–124