# Scroll, Match & Vote: An E2E Coercion Resistant Mobile Voting System

Gonçalo David Martins Tourais Pereira
goncalo.pereira@tecnico.ulisboa.pt

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa

## ABSTRACT

Mobile Internet Elections are appealing for several reasons: they promise voter convenience, lower abstention rates and lower costs. However, there are a number of trust issues that prevent them from becoming ubiquitous, the most relevant of which is the possibility of coercion of the voter at the time of the vote. But other issues, like the trustworthiness of both the services running the election and the mobile voting platform (usually the voter's computer or smartphone), are also major barriers to their adoption.

The proposed "Scroll, Match & Vote" (SM&V) system aims to overcome these trust issues, while ensuring the usability needed for wide adoption. SM&V builds upon previous e-voting solutions that ensure end-to-end verifiability and collusion resistance [13], and adds coercion resistance in a degree similar to the one of traditional booth voting systems.

SM&V requires the use of a device with Internet connection and a multitouch screen e.g. a smartphone. In the voting phase the voter is shown two lists side by side on the device. One of the lists contains all the election candidates and the other vote codes. One of these vote codes is correct the others are false. The voter votes by scrolling one or both lists and match her chosen candidate with the correct code.

## Keywords

Mobile Voting System, End-to-End Verifiability, Coercion Resistance

## 1. INTRODUCTION

Democracy is a form of government in which all electors have an equal say in the decisions that affect their lives. In our world most of the countries follow this system of government. The most common way of voting in a democratic country is made in controlled voting precincts, that must be made in a specific day.

One of the main problems of current western hemisphere general elections is the high abstention rate [3]. I argue that the space and time constraints of most elections increases that problem, and that by introducing internet voting these constraints may be mitigated.

Internet elections have been a research subject for many years leading to many interesting results. In spite of the risks to voter's privacy and election integrity, evidence seems to point out that Internet voting has come to stay. Accordingly with a 2007 study [15], numerous Internet elections ($\sim 139$) had already occurred worldwide and many of them ($\sim 40\%$) were actual real binding elections. These numbers have been increasing as more countries tried or adopted the Internet voting channel. A notable example is Estonia's case which is moving to/already have national binding Internet elections. A more recent example is Norway which ran an Internet voting system in 2011 [16].

The arguments in favor of Internet elections are obvious: i) increased voter convenience and participation, ii) tally accuracy and speed, iii) reduced costs, among others. However, the arguments against Internet elections are pertinent : i) the insecure voting platform problem, which results from the use of multipurpose devices owned and managed by the voter [21]; ii) the lack of transparency resulting from the nonexistence of physical votes and the possibility of collusion between the digital devices participating in the election; and iii) the nonexistence of private voting precincts that paves the way for several coercion scenarios.

The widespread use of smartphones with ubiquitous Internet access stressed, some of these advantages and disadvantages. While it is even more convenient for the voter to vote on her own smartphone, it is also easier for the coercer, given that the voter may vote anywhere. In spite of this, one of the most common reasons for failure of voting experiments is the lack of usability; voting systems that are too complex are deemed to fail, even if they are able to overcome all the above security disadvantages [19].

Scroll, Match & Vote (SM&V) is an end-to-end verifiable [17, 4] and collusion-resistant Internet voting system [13], with coercion resistant properties, although with a reduced mobility when compared with other Internet voting systems [13], given that part of the process must be performed in a controlled precinct.

Elections are usually constrained in time and space, i.e. they must be carried at the election-day and in controlled precincts. This double constraint is a source of abstention, given that not every voter is available to be at a specific place at that specific time. Removing either of these constraints is very problematic. If the election takes too long (i.e. several months) the democracy gets injured because some voters

vote with much less information than others. Early and postal-voting are seen as exceptions, not as rule. Removing the space-constraint is also difficult because it usually means loosing coercion resistance [5]. The current proposal follows the path of JCJ/Civitas [14, 5] and splits the two constraints such that the space-constraint and the time-constraint do not apply to the same action. The voter must register at a private booth without tight time constraints (within the timespan of one or two months) and must vote at election-day without any space constraints (with the exception of having Internet connection).

SM&V assumes that the voter owns a mobile Internet device with a multitouch screen (from now on referred as the voter's smartphone) with a secure element (either the UICC, an SDCard or an embed secure element) and proposes a new voting interface with coercion resistant properties that in combination with a variant of the EVIV voting protocol [13] attains also end-to-end verifiability and some resilience to collusion.

In the voting phase two lists of codes are shown to the voter side by side on the smartphone. One of the lists contains all the candidates and the other vote codes. One of these vote codes is correct (dubbed "pledge") the others are false. The voter votes by scrolling one or both lists and match her chosen candidate with the "pledge".

SM&V ensures integrity of the vote provided that $t$ out of $n$ configurable trustees are trustworthy. However, if the voter is coerced the protocol ensures confidentiality and integrity only if the secure element is trustworthy, thus in coercion scenarios the protocol is not collusion-resistant given that it only takes the secure element and the coercer to break integrity. Notice that this is a fundamental problem of end-to-end verifiable protocols; they ensure integrity only if the voter is able to verify the vote, if the voter is coerced not to verify they do not ensure integrity.

With SM&V, a voter may register several times and vote also several times, only the last one of both actions counts. The voter may also choose to register with SM&V and then choose another method to vote.

# 2. ARCHITECTURE AND TRUST MODEL

Scroll, Match & Vote (SM&V) is an end-to-end verifiable and $n$-collusion-resistant Internet voting system that attains also coercion resistance at the expense of reduced mobility when compared with EVIV.

SM&V proposes a new voting interface with coercion resistant properties combined with a variant of the EVIV voting protocol [13].

With the SM&V election system the voter may cast her vote anywhere using nothing more but her smartphone and an Internet connection, however, prior to election day, in a period that may be of a few minutes to a few months before the election, the voter must register for that election in a private booth. The privacy of this registration is paramount for the coercion resistant property of the voting system.

Besides the registration and voting phases there is yet another phase that every voter should do; the verification phase. The verification phase takes place any time after the voting phase with the purpose of ensuring that the casted vote is, in fact, counted as cast.

From a voter's perspective the voting machine is her smartphone, although as it is described below, the actual ballot creation is performed by an applet running inside a UICC, a secure SDCard or any other SE inside the phone.

# 3. ELEMENTS OF THE MODEL

Besides the voter and her smartphone, there are six other types of entities participating in the system.

The **Bulletin Board** ($\mathcal{BB}$) is the service responsible for the publication of all election public data. The data published cannot be deleted and it is always authenticated, i.e. digitally signed.

The **Trustee** service is run by $n$ different entities. The Trustee service exists in order to share the control over the voter's privacy and the election's integrity among several entities (the trustees). The trustees can be the political parties and/or any other authorized entity (e.g. an election observer or a non governmental organization).

The **Electoral Commission** (EC) service is responsible for the entire electoral process; namely, the EC is responsible for the voters enrollment system, the actual voting system and the authentication of all election public data.

The **Helper Organization** (HO) are entities that run services of vote verification by using public information published by the voting system. They can display to the voter her individual vote/receipt pairs. They also provide an application to generate the 2D-codes needed for the registration.

The **Pledge Display Device** (PDD) only purpose is build an untappable channel between the voter and the SE, to transport a small secret code to the voter: the "pledge".

Finally, the **Voting Machine** application that creates both the votes and the receipts is run by a secure element ($SE$) inside the smartphone of the voter, e.g. the UICC (Universal Integrated Circuit Card).

## 3.1 Properties and Trust Model

The proposed system exhibits a number of security properties under a specific number of assumptions. Some of the assumptions are necessary to ensure integrity-related properties, and others are needed to ensure confidentiality-related properties. Below, it is enumerated some of the most relevant properties and the assumptions under which they are achieved, and also some relevant limitations, i.e. relevant security properties that the proposed solution does not have. The assumptions of the system are the properties required from the environment where the system is to be deployed, therefore it is also discussed how they may be achieved and how reasonable they are.

### 3.1.1 Integrity Properties

$\mathbf{P}_{I1}$ - No votes can be added, deleted or modified without detection.

$\mathbf{P}_{I2}$ - Every vote is counted-as-recorded.

$\mathbf{P}_{I3}$ - Every voter can verify that her vote is recorded-as-intended with a soundness of $(1-2^{-\alpha})^{\rho\cdot(n_c-1)}$.[1]

These properties are ensured under the following assumptions:

$\mathbf{A}_{I1}$ - The data published in the $\mathcal{BB}$ cannot be deleted and it is always authenticated, i.e. digitally signed.

[1]The protocol security parameters $\rho$ and $\alpha$ are discussed in Section 4. $n_c$ is the number of running candidates.

$\mathbf{A}_{I2}$ - There is no collusion of more than $t < n$ out of $n$ trustees, where $t$ and $n$ are configurable security parameters.

$\mathbf{A}_{I3}$ - At least one honest organization or entity with cryptographic capabilities will verify the correctness of all the data published in $\mathcal{BB}$.

$\mathbf{A}_{I4}$ - The $SE$ and the coercer do not collude.

Assumption $A_{I1}$ is a common assumption of e-voting systems therefore some proposed web bulletin boards fit SM&V requirements [11]. The reasonableness of assumptions $A_{I2}$ and $A_{I3}$ depend on the number of available trustees and helper organizations. If there are enough trustees and helper organizations there is a high probability that there are no more than $t$ faulty trustees and there is at least one honest helper organization. Given that, SM&V trustees and helper organizations may be run by entities with no specific availability of performance requirements (cf. section 4), it is easy to assume that there will be a reasonable number of them available. Assumption $A_{I4}$ means that SM&V may not defeat a coercer which is able to simultaneously coerce the voter and tamper her secure element. This assumption is, nevertheless, much weaker than other coercion resistant assumptions. In JCJ/Civitas the whole vote machine is required to be trustworthy, independently that the voter is being coerced or not, while in SM&V only a small fraction of the voting machine is required to be trustworthy – the secure element, everything else, including the interface with the user and the communication with the voting machine do not need to be trusted.

#### 3.1.2 *Confidentiality properties*

$\mathbf{P}_{C1}$ - No one but the voter and her $SE$ knows the voter's chosen candidate.

$\mathbf{P}_{C2}$ - Coercion Resistant: Voters cannot prove how they voted, even if they can interact with the adversary while voting.

These properties are ensured under the following assumptions:

$\mathbf{A}_{C1}$ - The $SE$ (which performs the vote encryption) does not disclose the voters' vote choices.

$\mathbf{A}_{C2}$ - Neither the SE or the PDD disclose the "pledge" to anyone but the voter.

$\mathbf{A}_{C3}$ - Only legitimate registration precincts will possess certified PDDs, i.e. PDDs with a certificate signed by the election committee for that specific election with that specific validity.

$\mathbf{A}_{C4}$ - The channel between the PDD and the voter cannot be tapped.

Assumptions $A_{C1}$ and $A_{C2}$ requires trustworthy $SEs$. $SEs$ are secure by design. Although some exhibit vulnerabilities [18], it is far easier to build a trustworthy tailor-made $SE$ than a trustworthy tailor-made voting machine with all the user interfaces and communication devices. $A_{C2}$ requires that PDDs do not disclose the "pledges" to any one but the voter, which may be achieved, at some level, by removing any communication interfaces with the exception of the one used to communicate with the $SE$, e.g. NFC. Assumption $A_{C3}$ requires that PDD's certificates are issued with the identifier of the election for which they are being deployed and that each PDD is deployed on a valid registration booth. $A_{C4}$ is one of the most complex assumptions of the protocol to satisfy by the environment; any one with a camera is able to record and transmit what is being displayed by the PDD within the voting booth. However, this is a common assumption of most voting protocols, including the classical paper-based voting.

#### 3.1.3 *Relevant Non Properties*

**Force-Abstention** - An attacker may obtain a proof of abstention by looking at the tally and verifying if there is a vote for the coerced voter, Therefore anyone may force a voter to abstain and then verify if she complied. The voter may however vote physically without being detected by the coercer, given that any physical vote overrides e-votes.

**Randomization** - An attacker may force a voter to vote randomly, preventing the voter from voting on the chosen candidate. However, again, the voter may still vote physically given that any physical vote overrides e-votes.

**Pre-attack surveillance** - A coercer may learn with some probability the "pledge" of a voter by checking the $\mathcal{BB}$ and learning the code next to the probable candidate choice of the voter. After learning the "pledge" the coercer may force the voter to revote on another candidate. The coercer does not know, however, for sure, if the learned "pledge" is the correct "pledge". This vulnerability is shared with Civitas [5]. Again, in SM&V, the voter may evade the coercion by voting physically and replace her e-vote.

### 3.2 Scroll, Match and Vote

In a general view SM&V can be presented in figure 1. From a conceptual perspective the architecture in SM&V is similar to the EVIV system since it is a variant of it. The main differences in the process are described in the next.

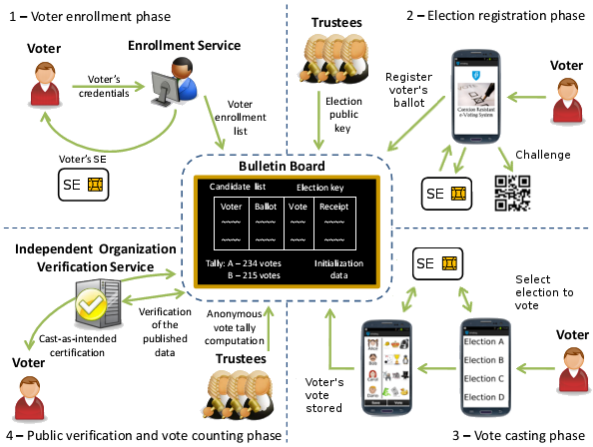


**Figure 1:** General view of SM&V.

### 3.2.1 Registration

The registration phase is the most complex phase of the voting process. It starts after a set of trustees generate the public key of the election and ends just before the vote casting, i.e. it can be made even on election day.

Prior to registration the voter should generate and print two 2D-codes and take it with her for the registration. The 2D-codes may be generated by the user, her self, by an online helper organization or even by a coercer, provided that he is not colluding with the SE (check assumptions in section 3.1). One of the 2D-codes (the second one to be used) contains a random number which is used to challenge the vote machine, and prevent it from generating a compromised ballot. The first 2D-code is a commitment to the second to prevent the voter from leaking the "pledge" to the coercer (cf. section 4.3). For usability purposes the two 2D-codes should be of different types (e.g. a PDF417 and a QR-code).

To register, the voter should take her smartphone to a private booth, specially prepared for the purpose, and press register on her smartphone voting application (Figure 2, step a). She will then be asked to: i) choose the election, ii) read one of the 2D-codes with her smartphone camera, iii) tap her phone against a special device dubbed "Pledge Display Device" (PDD), whose only purpose is build an untappable channel between the voter and the SE, to transport a small secret code to the voter: the "pledge".

The PDD owes its existence to the untrustworthiness of the voter's smartphone. Being a multipurpose device with many different running applications it is assumed that anything displayed on its screen may be leaked to a coercer. The PDD's only purpose is to receive, decrypt and display the "pledge". It does not know anything else about the voter therefore it cannot compromise the voter privacy. Still, to ensure that there is no possibility of using a false PDD to display the "pledge", only certified PDDs with a specific certificate signed by the electoral commission may be able to decrypt the "pledge" (cf. section 4.3).

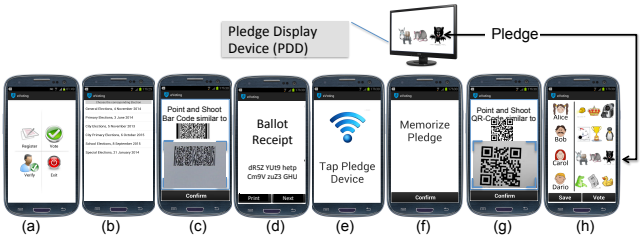


**Figure 2:** Registration procedure.

After tapping the smartphone on the PDD, the "pledge" appears on the PDD's screen and the voter is asked to memorize it. The pledge is a random number, generated by the vote machine, which is shown on the screen and encoded into a set of symbols. After confirming seeing the pledge, the voter is asked to read the second 2D-code using her smartphone's camera. Once the 2D-code is read, the smartphone displays the ballot to the voter. The ballot is shown as two scrollable lists side by side. The list on the left contains the names of the candidates, while the list on the right contains an equal number of sequences of symbols, one of which is the "pledge" shown in the PDD. To prevent coercion the voter should also memorize a few other sequences of symbols to be used as false "pledges" in case of coercion. The registration ends either by saving the generated ballot or by engaging immediately in the voting phase.

### 3.2.2 Voting

The voting phase starts with the same screen shown at the end of the registration. In fact, registration and voting may be done in a single sequence of steps at election day, or in two distinct sequences on separate occasions. Voting is accomplished by sliding one or both lists in the screen such that the chosen candidate and the sequence of symbols with the "pledge" become aligned (they can be visible or not, provided that they are aligned), and press VOTE. Anyone next to the user will not be able to tell in which candidate is the voter voting without knowing the "pledge". Given that the voter is able to lie to the coercer about the sequence encoding the "pledge", even a coercer will not be able to tell in which candidate the voter is voting.

### 3.2.3 Verification

After Voting the user should check if her vote was counted as intended by verifying that: her vote is in the poll, the 2D codes published match the printed ones, and that the vote is counted for the chosen candidate. The verification can be done using the mobile voting application, but it is recommended that the voter uses another Internet device with a simple web browser connected to a Helper Organization of her trust. Anyone with enough computer power might create a HO provided that it is trustworthy for some of the voters.

The task of a HO is to run a complete cryptographic check on all the votes and redo the vote tally to verify the overall result (cf. section 4.5). After verifying the correct construction of the ballot and vote the HO is able to display to the voter her individual vote/receipt pairs. The vote/receipt pair shown to the voter is similar to the voting screen shown to the voter, but without the ability to change the matching between the candidates and the sequences of symbols. The voter will then verify that the the code seen of the PDD display, i.e. the "pledge" appears next to her chosen candidate. If the voter verifies that her vote is not correctly cast, she may cast another vote on the Internet, or go to a vote precinct if that option is available.

## 4. SCROLL, MATCH AND VOTE (SM&V)

The SM&V election protocol has five phases: Voter Enrollment, Election Preparation, Ballot Registration, Vote Casting, and Vote Verification and Counting. The following sections describe the communication steps carried on each of these phases in detail. Each step is identified by an expression of the type $X \rightarrow Y : M$, where $X$ is the sender, $Y$ the receiver, $M$ the message, and $\rightarrow$ stands for either an NFC communication, an OTA communication or an interprocess communication within the smartphone. The expression $\langle M \rangle_{sk_B}$ stands for message $M$ signed by $B$, while $\langle M \rangle_{pk_B}$ stands for message $M$ encrypted for $B$ and $\{n_j\}_{j=1}^{k}$ stands for a set with every element $n_1$ to $n_k$. It is assumed that the voter's smartphone is able to communicate through NFC with the PDD. NFC communication starts only when two NFC-enabled devices get almost in touch with each other (dubbed tap each other) and, at least, one of them had queued a message to be sent by NFC.

## 4.1 Voter Enrollment

Voter enrollment takes place only once per voter and is valid for several elections.

1. $\mathcal{V}_i \rightarrow EC$ : Voter Credentials

   Voter enrollment starts when the voter $\mathcal{V}_i$ provides the Electoral Commission her credentials proving to be a valid voter.

2. - $EC \rightarrow \mathcal{V}_i$ : SDCard with Voting Machine and the Voter Electoral Credentials
   - $EC \rightarrow TSM \rightarrow UICC_i$ : Upload Voting Machine and the Voter Electoral Credentials

   In response, the Electoral Commission either provides the voter with a secure SDCard containing the voting machine and the voter credentials to be used for voting, or uploads the voting machine software to the voter's UICC, along with the voter electronic electoral credentials, through a OTA Trusted Service Manager (TSM) [10]. The former option is simpler and more secure but requires delivering a physical card to each voter.

## 4.2 Election Preparation

The election preparation phase takes place once per election and is performed by the Electoral Commission, the Election Trustees and the $\mathcal{BB}$.

1. $EC \rightarrow \mathcal{BB} : \langle electionParameters, \mathcal{C} \rangle_{sk_{EC}}$

   The election preparation phase starts with the Electoral Commission publishing on the Bulletin Board the election candidate list $\mathcal{C}$ and the public election parameters, such as: the election date and the election security parameters (e.g. election key pair parameters).

2. $\mathcal{T} \rightarrow \mathcal{BB} : \langle keyGenerationData,\ pk_{\mathcal{T}} \rangle_{sk_{\mathcal{T}}}$

   The second step in the election registration phase is the creation of a shared threshold ElGamal election key pair by the set of Trustees $\mathcal{T}$. In [20, 9] the reader can find more details on how to create a $(t, n)$-threshold election key pair, for the ElGamal cryptosystem, and how to decrypt a message using the shared private key. The input messages (cryptographic key parameters), the public outputs of the key generation protocol and the election public key ($pk_{\mathcal{T}}$) are all published in the public Bulletin Board. Each trustee signs her messages before sending them to the Bulletin Board.

3. $EC \rightarrow \mathcal{BB} : \langle id_{election}, pk_{\mathcal{T}}, \mathcal{C} \rangle_{sk_{EC}}$

   The Electoral Commission verifies the election public key generation data, published by the Trustees, and validates it by signing together with the candidate list $\mathcal{C}$ and the election identifier (which is a generic identifier), and publishing the signed tuple on the Bulletin Board.

## 4.3 Ballot Registration

Each voter may register one or several ballots for each election, although only the last one may be used for voting. There are five entities involved in the registration phase: the voter, the voter's smartphone, the SE, the PDD, and the $\mathcal{BB}$ (Figure 3).

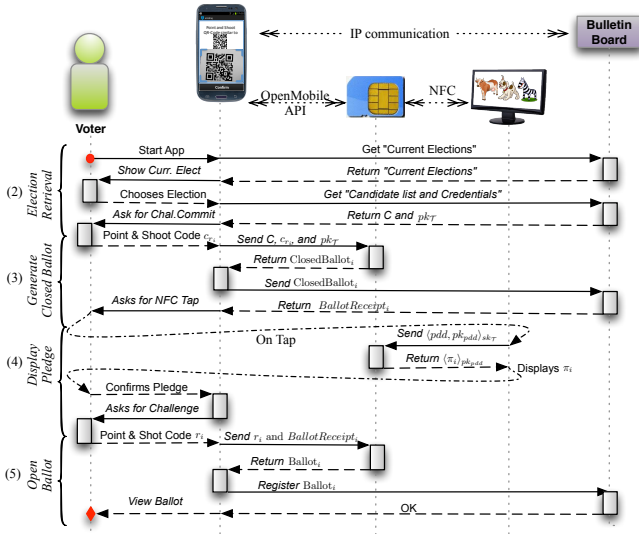


**Figure 3:** Registration procedure.

The registration process starts outside the voting booth with the generation of the 2D-codes and ends inside a private booth. The registration has 5 distinct stages:

(1) 2D-code generation;

(2) Election parameters retrieval;

(3) Closed Ballot Generation;

(4) Pledge Display;

(5) Partial Opening of the Ballot.

The first two stages are just preliminary stages with the intent of gathering information about the election. In the third stage the voting machine (SE) generates and commits to a closed ballot, i.e. a ballot that must be open before being used to vote. In the fourth stage the voting machine displays the "pledge" to the voter, which will be used has a partial key to open the ballot. Finally, in the fifth stage the voter challenges the vote machine with one of the 2D-codes and the voting machine replies with a partial key for the ballot. Only someone with the "pledge" and this partial key is able to use the ballot to vote. Each of these stages is idempotent and starts with the voter and ends with the voter, therefore it may be repeated several times in case of a communication or smartphone failure (e.g. battery depletion). It follows a description of the steps in each of these stages:

1. 2-D Code Generation

   (a) $\mathcal{V}_i$ or $HO \rightarrow \mathcal{V}_i$ : QR-code$(r_i), PDF417(c_{r_i})$

   The registration starts when each voter $\mathcal{V}_i$ generates, either with the help of an online HO or on her own, one random number $r_i \in_R \mathbb{Z}_q$ and one commitment $c_{r_i} = H(r_i)$, and print the former in a QR-code and the latter in a PDF417 code, which are used in stages (3a) and (5), respectively.

2. Election parameters retrieval

(a) $\mathcal{BB} \rightarrow$ Smartphone $\rightarrow SE : \langle id_{election}, pk_{\mathcal{T}}, \mathcal{C} \rangle_{sk_{EC}}$
After generating the two 2D-codes, the voter requests, from the $\mathcal{BB}$, the public key $pk_{\mathcal{T}}$ and the candidate list $\mathcal{C}$ of the election, signed by the electoral commission, and forwards it to the secure element.

3. Closed Ballot Generation

(a) $PFD417(c_{r_i}) \rightarrow$ Smartphone $\rightarrow SE : c_{r_i}$
The third stage starts by asking the voter to point her smartphone camera to the PDF417 code and read the commitment $c_{r_i} = H(r_i)$ encoded in it, which is also forwarded to the SE. This step prevents the voter from establishing a covert channel using the random number $r_i$ to leak the "pledge" in stage (5).

(b) $SE \rightarrow$ Smartphone $\rightarrow \mathcal{BB} : \langle c_{r_i}, ClosedBallot_i \rangle_{sk_i}$

The SE element replies by generating a closed ballot and forwards it to the $\mathcal{BB}$, through the smartphone. This ballot is similar to an MP3 vote [12] but for a random candidate, i.e. the ballot contains already a full fledged vote but no one knows for which candidate until the ballot is partial open in stage (5). The ballot is comprised by $n_c$ tuples, one for each candidate, with two exponential ElGamal encryptions, each.

$$ClosedBallot_i = \{cv_{ij}\}_{j=1}^{n_c} = \{\mathcal{E}_{pk_{\mathcal{T}}}(m_{ij}, \tau_{ij}), \mathcal{E}_{pk_{\mathcal{T}}}(\theta_{ij}, \delta_{ij})\}_{j=1}^{n_c}$$

The first encryption contains $m_{ij} = 1$ for a random candidate $j = p_i \in_R \mathbb{Z}_{n_c}$ (i.e. a YESVote for a random candidate $p_i$) and $m_{ij} = -1$ for all the others. The second encryption, encrypts a random number $\theta_{ij} \in_R \mathbb{Z}_q$, where $\pi_i = \theta_{ij}|_{j=p_i}$ is dubbed "pledge" of voter's $\mathcal{V}_i$. $\tau_{ij} \in_R \mathbb{Z}_q$ and $\delta_{ij} \in_R \mathbb{Z}_q$ stand for the usual ElGamal secret randomization factors.

The SE also generates and publishes along with the closed ballot a couple of proofs that $\forall j : m_{ij} \in \{1, -1\}$ and that $\forall i : \exists! j : m_{ij} = 1$, respectively, which are omitted for brevity, and modeled by two oracles $\Omega(v)$ and $\Omega(cv)$, respectively. Check [12, 8, 7] for details.

(c) $\mathcal{BB} \rightarrow$ Smartphone $\rightarrow SE : BallotReceipt_i = H(\langle c_{r_i}, ClosedBallot_i \rangle_{sk_{\mathcal{BB}}})$
The $\mathcal{BB}$ replies with a signed receipt for the ballot – $BallotReceipt_i$, which is also shown to the user and optionally printed.

4. Pledge Display

The "pledge" display is the first stage that must and should be done within the controlled precinct. It starts with a tap between the voter's mobile phone and the PDD. The tap marks the starting of the process that must be done inside a private booth.

(a) $PDD \rightarrow SE : \langle PDD, id_{election}, pk_{pdd} \rangle_{sk_{EC}}$
After detecting the tap the PDD sends its own certificate to the SE inside the mobile phone. The SE checks the signature on the certificate and the election identifier to be sure that the "pledge" will be sent to an official PDD.

(b) $SE \rightarrow PDD \rightarrow \mathcal{V}_i : \langle \pi_i \rangle_{pk_{pdd}}$
The SE takes the public key $pk_{pdd}$ of the PDD's certificate, encrypts the "pledge" $\pi_i$ with it, and sends it back to the PDD on the same tap. The PDD decrypts the "pledge", takes the least $\alpha$ significant bits ($\pi \bmod 2^{\alpha-1}$), converts it to a sequence of symbols and displays them on its screen. When the voter sees the "pledge" on the PDD's screen she presses the confirmation button on her smartphone

5. Partial Opening of the Ballot

The fifth stage starts by asking the voter to point her smartphone camera to the previously generated QR-Code and proceed as follows:

(a) QR-code$(r_i) \rightarrow$ Smartphone $\rightarrow SE : r_i$
The number read from the QR-code $r_i$ is sent to the SE which checks that it matches the commit-value previously read from the PDF417 code, i.e. $c_{r_i} \stackrel{?}{=} H(r_i)$.

(b) $SE \rightarrow$ Smartphone $\rightarrow \mathcal{BB} : Ballot_i$
If both numbers match, the SE generates a $PartialKey_i$ and builds a complete ballot with it and the previous generated items.

$$Ballot_i = \langle \mathcal{V}_i, CB_i, BR_i, PK_i, chal_i^n, c_{r_i}, r_i \rangle_{sk_i}$$ [2]

The $PartialKey_i$ is comprised by $n_c$ tuples, one for each tuple in $ClosedBallot_i$.

$$PartialKey_i = \{\vartheta_{ij}, \omega_{ij}\}_{j=1}^{n_c}$$

The first element $\vartheta_{ij}$ of each of the tuples is either the "pledge" $\pi_i$, or some hidden but provable correct value, resulting from the $\theta_{ij}$ and some challenge $chal_i^n$.

$$\vartheta_{ij} = \begin{cases} \pi_i & \text{if } j = p_i (YESvote) \\ 2 \cdot chal_i^n - \theta_{ij} \bmod q & \text{if } j \neq p_i (NOvote) \end{cases}$$

The challenge $chal_i^n$ must be fresh and unpredictable by the secure element, otherwise the secure element could guess $\vartheta_{ij}|_{j \neq p_i}$ before displaying the "pledge" to the voter and fool the voter by showing a different $\vartheta_{ij}$ as "pledge".

$$chal_i^n = H(r_i, BallotReceipt_i, n) \bmod q$$

$$n \in_R \{1, \ldots, \rho\}$$

The unpredictability of the challenge is ensured by the random value $r_i$ encode in the QR-Code under assumption $A_{I3}$, and the freshness by the $BallotReceipt_i$. For reasons of usability, only $\alpha$ bits of $\vartheta_{ij}$ are shown to the user in the next step, therefore it is not unlikely that two or more of these codes would be perceived has equal by the voter. To prevent that, the SE element generates

[2] $CB = ClosedBallot$, $BR = BallotReceipt$, $PK = PartialKey$

at most $\rho$ different challenges until a $PartialKey_i$ with no perceived duplicates is generated [13].

The second element $\omega_{ij}$ of each of the tuples is a proof of correct generation of the first element $\vartheta_{ij}$ (check step (1) in section 4.5).

$$\omega_{ij} = \tau_{ij} \cdot (chal_i^n - \vartheta_{ij}) + \delta_{ij} \bmod q$$

(c) Smartphone $\rightarrow \mathcal{V}_i : \{\vartheta_{ij} \bmod 2^{\alpha-1}\}_{j=1}^{n_c}$
Finally, the least $\alpha$ significant bits of each $\vartheta_{ij}$ are sent to the smartphone and displayed to the voter encoded in a sequence of symbols. Only then the voter is allowed to leave the booth.

## 4.4 Voting

The voting phase starts either immediately after the registration ends, using the end screen of the registration, or when the voter runs again the SM&V application on her smartphone and asks the SE for unused ballots.

1. $\mathcal{V}_i \rightarrow$ Smartphone $\rightarrow SE : rot_i$

   The voter takes the ballot $Ballot_i$ and the official list of candidates and rotates the ballot until the entry of the $YESvote$ matches the chosen candidate. The voter is the only one that knows which entry of the ballot contains the $YESvote$ because it is the only one that knows the "pledge". The chosen rotation is then forward to the secure element.

2. $SE \rightarrow$ Smartphone $\rightarrow \mathcal{BB} : Vote_i$

   The secure element takes the rotation value, signs it together with a digest of the ballot and forwards to the $\mathcal{BB}$,

   $$Vote_i = \langle \mathcal{V}_i, rot_i, H(Ballot_i) \rangle_{sk_i}$$

   which stores it along with the rest of the voter's data

   $$\mathcal{BB}_i = \mathcal{V}_i, Time_i, BallotReceipt_i, Ballot_i, Vote_i$$

   The voter may vote any number of times, only the last one counts. Therefore, if some communication or smartphone failure occurs in the process, the voter may just repeat the failing step, or the complete voting process.

## 4.5 Tally & Verification

The tally and verification phase is similar to the EVIV's [13] tally and verification phase. It is comprised by four steps: a preliminary verification step, a deduplication step, a user verification step and a counting step.

1. The preliminary verification step can be carried as soon as each vote arrives to the $\mathcal{BB}$. It consists of verifying:

   i) the signatures of each element in the $\mathcal{BB}$: $BallotReceipt_i$, $Ballot_i$ and $Vote_i$;

   ii) the casted vote is for the registered ballot i.e. the digest $H(Ballot_i)$ within $Vote_i$ matches the digest of ballot $Ballot_i$;

   iii) the challenge is correctly generated i.e. $chal_i^n \stackrel{?}{=} H(r_i, BallotReceipt_i, n) \bmod q, \quad 0 < n \leq \rho$ and $c_{r_i} \stackrel{?}{=} H(r_i)$).

   iv) all encrypted values $m_{ij}$, within $Ballot_i$, are either 1 or $-1$, which may be achieved using a Zero-Knowledge Proof such as the one proposed in [12, 8, 7], and is modeled here by Oracle $\Omega(cv)$.

   v) only one of the $m_{ij}$ is 1, which may be achieved using a Zero-Knowledge Proof such as the one proposed in [12], and is modeled here by Oracle $\Omega(v)$.

   vi) the $PartialKey_i$ used to open the ballot was correctly generated by checking the Zero Knowledge proof [12]:

   $$\mathcal{E}_{pk_\mathcal{T}}(chal_i^n, \omega_{ij}) \stackrel{?}{=}$$

   $$(\mathcal{E}_{pk_\mathcal{T}}(m_{ij}, \tau_{ij}))^{chal_i^n - \vartheta_{ij}} \cdot \mathcal{E}_{pk_\mathcal{T}}(\theta_{ij}, \delta_{ij})$$

   Every non-complying vote is removed from the tally.

2. As the name implies the deduplication step consists on the removal of vote duplicates by checking the time of their arrival. Only the last vote for each voter should remain in the tally. All valid votes and their receipts should be published in the $\mathcal{BB}$.

3. The voter verification step is carried by the voter over the published list of valid votes. The voter is able to check that her vote is cast as intended by checking that the "pledge" in her published vote receipt is positioned by the side of her chosen candidate, that the $r_i$ in the QR-code used to generate the challenge matches the one used at the time of the voting, and that the ballot receipt is the one presented to the voter.

4. Finally, the actual counting is performed by either using a mix net or using the homomorphic properties of exponential ElGamal. Using the homomorphic properties the result of the tally $d_j$ for each candidate $j$ can be computed independently by performing the homomorphic addition of each candidate vote $\mathcal{E}_{pk\mathcal{T}}(m_{ij}, \tau_{ij})$ of each vote $Vote_i$. Given that $m_{ij}$ is either 1 or $-1$ the homomorphic addition is

   $$\bigoplus_{i=1}^{n_v} \mathcal{E}_{pk_\mathcal{T}}(m_{ij}, \tau_{ij}) = \mathcal{E}_{pk_\mathcal{T}}\left(\frac{n + d_j}{2}, \varphi\right)$$

   Which is decrypted by the election trustees using their shared secret key $sk_\mathcal{T}$ and solved in order to $d_j$, which is then publish in the $\mathcal{BB}$ along with a proof of correct decryption (cf. [8]).

Only the last step must be done at the end of the election all the remaining ones may, and should, be done as soon as the vote arrives to the $\mathcal{BB}$.

## 4.6 Post Election Audit

The Post Election Audit is carried, independently, by several HO and mimics the Tally&Verification phase. The only difference is the actual decryption of the tally result. Instead of feeding the encrypted result to the trustees to be decrypted, HOs compare their encrypted result with the one computed in the Tally&Verification phase, and check the proof of correct decryption provided by the trustees in the Tally&Verification phase.

# 5. EVALUATION

The evaluation of this project is very important to be examine. This section explains the usability options and their impact, the performances achieved with the SM&V prototype and finally a general discussion of security of the protocol is done.

## 5.1 Usability

Usability is a major issue in any voting system, but assumes a specific relevance in end-to-end voting systems, where the voter distrusts her voting machine and is, therefore, required to handle a complex voting interface.

SM&V requires the voter to be able to memorize the "pledge" for a long period (sometimes over a month) and be able to distinguish it from the remaining of the verification codes. Both the "pledge" $\pi_i$ and the verification codes $\vartheta_{ij}$ are large random numbers in $\mathbb{Z}_q$, not suitable for memorization or display to the voter. To be displayable, these numbers are truncated to $\alpha$ bits, by applying the $\mod 2^\alpha$ operation (see section 4.3), and coded into a sequence of different symbols. The size of the sequence of symbols depends both on the value of $\alpha$ and on the number of different symbols. A small sequence of symbols simplifies memorization, but implies a larger set of symbols, which complicates distinguishability, therefore, it is expected that choosing the correct set of symbols may have a significant impact on the overall usability of the system.

According with Bertin [2] there are 8 visual variables that are used by humans to distinguish symbols: shape, size, color, brightness, pattern, orientation and horizontal and vertical position. Symbols that differ in more variables are easer distinguishable from each other; therefore it is possible to use large sets of symbols provided that they differ in as many as these variables as possible. On the other hand, long term memory of humans works better with semantic information [6] rather than abstract information, which seems to point that symbols representing concrete concepts are preferred over abstract ones.

The quality of the chosen set of symbols was tested by an experiment with 45 different subjects, with the distribution of age, gender and education level according with Tables 1 and 2. To each of the subjects it was shown a sequence of three symbols similar to the "pledge" and a list of sequences of three symbols similar to the ballot. It was then asked the subjects to find the "pledge" in the ballot and memorize both the "pledge" and the position where it appears in the ballot. A copy of the ballot was given to the subjects, who were also instructed not to make any mark or written annotation about the "pledge". Finally, a month later, the subjects were asked to point the "pledge" in the ballot.

The results are very promising, although there is still some margin for improvement. Only 3 of the 45 subjects (6.7%) were not able to point the "pledge" within the ballot, resulting into $93.3\% \pm 6\%$ correctness for a confidence level of 0.9. The reasons for these errors were completely transversal to gender, age or education level. From the three subjects that forgot the pledge, two made a confusion about two of the symbols that were too much alike, and the other mistakenly identified a code similar to the "pledge" of a previous experiment. These two types of mistakes confirmed the relevance of a good choice of symbols (they should be very different from each other), and revealed that consecutive elections should not share the same set of symbols. Both problems may be solved easily.

| | Gender | | | |
|---|---|---|---|---|
| Age | Male | | Female | |
| 15-24 | 5 | (11.4%) | 8 | (18.2%) |
| 25-49 | 22 | (50%) | 6 | (13.6%) |
| 50-64 | 2 | (4.5%) | 1 | (2.3%) |
| $> 64$ | 0 | (0%) | 0 | (0%) |

**Table 1:** Distribution of subjects by age and gender

| Education Level | Percentage |
|---|---|
| Basic Education | 18.5% |
| Secondary Education | 14.8% |
| College Education | 66.7% |

**Table 2:** Distribution of subjects by education level

| Memo technique | Number | |
|---|---|---|
| Sequence of symbols of the "pledge" | 15 | (29.4%) |
| Non repeating symbol of the "pledge" | 12 | (23.5%) |
| Candidate in front of the "pledge" | 8 | (15.6%) |
| "Pledge" position within the ballot | 7 | (13.7%) |
| History with the symbols of the "pledge" | 3 | (5.88%) |
| Other | 6 | (11.8%) |

**Table 3:** Memorization techniques reported by the voters

Nevertheless, voters that forget the "pledge" or are uncertain of it, may register again and receive another "pledge", or may even decide to invalidate their Internet registration and vote using the classical way or any other voting method.

The voters that chose the correct "pledge" reported several techniques to memorize it (Table 3). While some reported to have memorized all three symbols in the "pledge" (29.4%), others memorize just one symbol that they found not to repeat in another position of the ballot (23.5%). Others yet, memorized the candidate which was in front of the "pledge" when the ballot was saved (15.6%). Finally, some memorized the position of the "pledge" in the ballot (13.7%). Note that some voters used several memorization techniques.

Another interesting result was the perception of difficulty of the task; the task is perceived to be much more difficult then it is. While 28.9% of the subjects stated, in the beginning of the experience, that they were expecting to fail (i.e. forgetting the "pledge"), the reality is that only 6.7% (3 subjects) did it. This error in the perception of the difficulty of the task may result from modesty, i.e. the voter may not want to brag about her ability to memorize the code without testing how difficult it is. But it may also result from not perceiving correctly the task being asked. In fact, several voters showed surprised when they were told that they may keep the "pledge" written in the ballot together with the other codes and just have to memorize which of them it is the "pledge", and may even refresh their memory from time to time, if they want.

## 5.2 Performance

From a performance point of view, the voting machine is, also, the critical element, and the registration phase the most critical phase, because it is the one where most cryptographic operations are performed. Recall, that the voting phase requires no more than a signature over a digest of the

ballot and a small rotation number. Table 4 shows the delay experienced by the voter at the registration phase. The table shows the delays before each screen shown to the user (cf. Figure 2), for a 10-candidate election, and the time, per candidate, taken by each operation contributing for those delays. All the measurements were taken with $p = 1536$ bits and $q = 1024$ bits.

The biggest delay is between point & shoot the PDF417 code, with the commitment to the challenge, and the screen asking the voter to tap the PDD (screens c $\rightarrow$ d), which is around 4.2 min. However, this is not the most critical delay given that it may be performed before entering the controlled precinct. The most critical delay is the last one; after point & shoot the QR-code, with the challenge, and being presented with the ballot to vote (screens g $\rightarrow$ h), which is around 12 sec. The whole process takes $\approx 4.4$ min, although the time that must be spent within the controlled precinct is under 15 sec, which is reasonable, although it may be further improved with code optimizations and a card featuring modular multiplication [12].

## 5.3 Security Discussion

The proposed system is resistant to coercion attacks only if the "pledge" is not known by anyone but the voter, and the voter becomes aware of the "pledge" at the same time that it becomes aware of the false "pledges", i.e. the verification codes in the receipt. Which means that the channel between the SE and the voter should not be tappable or interruptible, i.e. the voter should only be allowed to leave the controlled precinct after receiving the receipt, which can be achieved if the last vote registration message may only be sent through the secure environment network. With this assumption the voter may always point to a different code within the receipt, when asked by a coercer, without risk of being caught. Nevertheless, a coercer is still able to force a voter to vote randomly and prevent her from voting.

The system is resistant to a configurable degree of collusion among the participating entities. In fact, the bulletin board, the helper organizations and the electoral commission only handle public information, therefore as long as one HO is honest to help the voter verifying her vote, all the rest may collude without any consequences. However, the number of faulty trustees must be less than $t$, otherwise the private key of the election is compromised and every vote can be decrypted. This degree of collusion resistance leaves the system less vulnerable against cross-infection between the elements of the model, however some care must be taken with the SE and the PDD. Both elements are in contact with the smartphone, which is assumed to be infected, and if either of these elements gets infected it could compromise the confidentiality of the vote. SM&V relies on the secure design of the secure element and on the simplicity of the PDD to make them immune to smartphone infections.

Nevertheless, even if either the SE, the PDD, or both get infected only the confidentiality of the vote gets affected. If the challenge is correctly generated the secure element can only trick the voter with a probability $1 - p_{soundness}$, $p_{soundness} = (1 - 2^{-\alpha})^{\rho \cdot (n_c - 1)}$, which can be set to a configurable low value. Similarly, the probability that the PDD is able to trick the voter in voting for a different candidate is $p_{pdd} = 1 - (1 - 2^{-\alpha})^{(n_c - 1)}$, which is the probability that the PDD is able to guess one of the other verification codes.

The system is not resistant to collusion between a coercer and the SE. In fact, if the coercer is able to prevent a voter from complaining about a wrongly casted vote and the SE is compromised, it can submit a vote on any candidate without being detected. Secure elements are designed to be secure, however some are more secure than others. The SE used for the SM&V prototype was a GO-Trust secure microSD card running Global Platform 2.2.2 and Java 2.2.2 applets.

The solution requires the use of two 2D-codes. The first 2D-code is a commitment for the second 2D-code that encodes the challenge. The goal of the first 2D-code is to prevent a covert channel between the vote and the coercer through the challenge [1]. The 2D-codes may be generated by anyone, including the coercer, provided that it is not colluding with the SE. If the entity generating the 2D-codes and the SE collude it could be possible to change the order of the challenge and commitment to the challenge, which would allow the SE to elude the voter.

# 6. CONCLUSIONS

The Internet voting systems are very appealing for the society. Studies shows that these systems are increasing in adoption all over the world [15].

Although, secure mobile Internet elections is a hard goal to achieve, and there is still a long way until all relevant properties are attained simultaneously, I believe SM&V is a step in that direction, mainly because, to my knowledge, it is the first to ensure, simultaneously, a set of security properties (section 3.1) in the remote voting context, without relevant usability degradation.

SM&V assumes that the voter owns a mobile Internet device with a multitouch screen with a secure element (either the UICC, an SDCard or an embed secure element) and proposes a new voting interface with coercion resistant properties that in combination with a variant of the EVIV voting protocol [13] attains also end-to-end verifiability and some resilience to collusion.

In the voting phase the voter is shown two lists side by side on the device. One of the lists contains all the election candidates and the other vote codes. One of these vote codes is correct the others are false. The voter votes by scrolling one or both lists and match her chosen candidate with the correct code.

The proposed system is resistant to coercion attacks only if the "pledge" is not known by anyone but the voter, and the voter becomes aware of the "pledge" at the same time that it becomes aware of the false "pledges". With this assumption the voter may always point to a different code within the receipt, when asked by a coercer, without risk of being caught. Nevertheless, a coercer is still able to force a voter to vote randomly and prevent her from voting.

In the verification step the voter is able to check that her vote is cast as intended by checking that the "pledge" in her published vote receipt is positioned by the side of her chosen candidate, that the QR-code matches the one used at the time of the voting, and that the ballot receipt is the one presented to the voter.

The SM&V protocol in comparison to EVIV adds the element PDD. The PDD owes its existence to the untrustworthiness of the voter's smartphone. Being a multipurpose device with many different running applications it is assumed that anything displayed on its screen may be leaked to a coercer. The PDD's only purpose is to receive, decrypt and display the "pledge" to the voter.

| Registration Screens (Figure 2) | Delay between screens ($n_c = 10$) | Registration steps (Section 4.3) | Actions performed between screens | Time (s) |
|---|---|---|---|---|
| b) → c) | 0.2 s | (2) | Election parameters retrieval | 0.2 |
| c) → d) | 4.2 min | (3a) | Set challenge commit | 0.04 |
| | | (3b) | Create closed ballot | $8n_c + 0.9$ |
| | | | Create ZK proofs | $16.2n_c$ |
| | | | Get closed ballot from SDCard | $0.4n_c$ |
| | | | Get ZK proofs from SDCard | $0.3n_c$ |
| e) → f) | 0.04 s | (4) | Pledge display | 0.04 |
| g) → h) | 12.5 s | (5a) | Set challenge | 0.04 |
| | | (5b) | Create ballot | $1.2n_c + 0.6$ |
| | | | Get ballot from SDCard | $0.04n_c$ |
| | 4.4 min | | | |

**Table 4:** Performance times for the registration phase

## 7. ACKNOWLEDGMENTS

I want to thank my advisor Prof. Carlos Ribeiro who gave me unconditional support in all phases of the thesis. I would also like to acknowledge the help and the availability Rui Joaquim and André Brioso gave me in clarifying the doubts which have arisen and which were answered promptly.

## 8. REFERENCES

[1] B. Adida and C. A. Neff. Ballot casting assurance. In *EVT 2006*, pages 7–15, Berkeley, CA, USA, 2006. USENIX Association.

[2] J. Bertin. *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin press, Wisconsin, 1983.

[3] J. Brittain. Clandestine politics and bottom-up organizing in colombia. *Journal of Latin American Studies (2005)*.

[4] D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy*, 2:38–47, 2004.

[5] M. Clarkson, S. Chong, and A. Myers. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy*, pages 354–368, Oakland, CA, USA, May 2008. IEEE Computer Society.

[6] F. I. Craik and R. S. Lockhart. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior*, 11(6):671–684, 1972.

[7] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94*, volume 839 of *LNCS*, pages 174–187, Santa Barbara, CA, USA, 1994. Springer.

[8] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *EUROCRYPT '97*, volume 1233 of *LNCS*, pages 103–118. Springer Berlin / Heidelberg, Konstanz, Germany, 1997.

[9] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Advances in Cryptology—CRYPTO'89 Proceedings*, pages 307–315, Santa Barbara, CA, USA, 1990. Springer.

[10] Global Platform. Card specification v2. 1.1. Technical report, Online: http://www. globalplatform. org, 2003.

[11] J. Heather and D. Lundin. The append-only web bulletin board. In *Formal Aspects in Security and Trust*, pages 242–256. Springer, Eindhoven, The Netherlands, November 2009.

[12] R. Joaquim and C. Ribeiro. An efficient and highly sound voter verification technique and its implementation. In A. Kiayias and H. Lipmaa, editors, *E-Voting and Identity*, volume 7187 of *Lecture Notes in Computer Science*, pages 104–121. Springer Berlin / Heidelberg, Tallinn, Estonia, 2012.

[13] R. Joaquim, C. Ribeiro, and P. Ferreira. Eviv: an end-to-end verifiable internet voting system. *Computers & Security*, 32:170–191, 2012.

[14] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *Proc. of the 2005 ACM workshop on Privacy in the electronic society*, pages 61–70, Alexandria, VA, USA, November 2005. ACM.

[15] R. Krimmer, S. Triessnig, and M. Volkamer. The development of remote e-voting around the world: A review of roads and directions. In *E-Voting and Identity*, volume 4896 of *LNCS*, pages 1–15. Springer Berlin / Heidelberg, Bochum, Germany, october edition, 2007.

[16] Ministry of Local Government and Regional Development. e-vote 2011 - project web site, September 2012. `http://www.regjeringen.no/en/dep/krd/prosjekter/e-vote-2011-project.html?id=597658`.

[17] C. A. Neff. Practical high certainty intent verification for encrypted votes. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.134.1006`, 2004. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.134.1006`.

[18] K. Nohl. Rooting sim cards. In *BlackHat US 2013*, Las Vegas, NV, July 2013.

[19] A.-M. Oostveen and P. Van den Besselaar. Security as belief: user's perceptions on the security of electronic voting systems. *Electronic voting in Europe: Technology, law, politics and society*, 47:73–82, 2004.

[20] T. P. Pedersen. A threshold cryptosystem without a trusted party. In *Advances in Cryptology—EUROCRYPT'91*, pages 522–526, Brighton, UK, April 1991. Springer.

[21] A. D. Rubin. Security considerations for remote electronic voting. *Commun. ACM*, 45(12):39–44, Dec. 2002.