Chapter 16

Electronic Voting Systems

16.1 INTRODUCTION

For several years now the identification of the user requirements that an electronic voting system should satisfy attracts the interest of both governments and research communities. The main difficulty of the requirements elicitation process seems to be the different perspective of each side: governments refer to requirements as the set of applicable laws pertaining a certain voting procedure, while researches don't go much further than simply providing a narrative description of system's non-functional characteristics related to security. Both sides seem to underestimate the fact that an electronic voting system is an information system with functional, as well as non-functional, requirements.

Functional requirements may vary from one system to the other since they depend on the needs of the market segment that the system will serve. However, this is not the case for the vast majority of security requirements. They are similar to all e-voting systems since they aim to ensure compliance of the system with the election principles and the security and privacy issues dictated by the international legal frameworks. Security requirements are, at a large extent, fulfilled by the voting protocol adopted by the system.

The first part of this chapter includes the complete list of functional and non-functional requirements for an electronic voting system, taking into account the European Union legislation, the organizational details of currently applicable voting procedures and the possibilities offered, as well as the constraints imposed, by the latest technology. Following that, there is a detailed presentation of several generic and enhanced models, proposed in the cryptographic literature, for remote e-voting, as well as of a new class of cryptographic voting schemes for paper-based elections in polling stations.

16.2 REQUIREMENTS FOR AN INTERNET-BASED E-VOTING SYSTEM

The decision to build an electronic voting system in order to conduct elections over public networks (i.e. Internet) is neither an easy nor a straightforward one. The reason being that a long list of legal, societal and technological requirements must be fulfilled [48][35]. A further difficulty is that a vast majority of the system requirements has been produced by transforming abstract formulations -- i.e. laws or principles like “preserve democracy”-- to a concrete set of functional and non-functional requirements.

The **functional requirements** of an e-voting system specify, in a well-structured way, the minimum set of services (tasks) that the system is expected to support, highlighting at the same time their desired sequence and all possible interdependencies. For instance, the number and type of elections processes (e.g. polls, referendums, internal elections, general elections etc) supported by an e-voting system are determined by its set of functional requirements. Furthermore, functional requirements are related to many of the usability properties of the system, dominating the properties and characteristics of its interaction model with the user. On the other hand, **non-functional requirements** are related to the underlying system structure, in principle they are invisible to the user and they normally have a severe impact on architectural decisions. Security requirements and several system wide properties like flexibility, voter convenience, efficiency etc, are derived through the set of non-functional requirements.

16.2.1 Functional Requirements

In principle, functional requirements for e-voting systems may vary a lot, since each system is aiming to fulfil the specific requirements of the market segment that it is targeting. However, the most common objectives of an e-voting system are to [35]:

1. Provide the entire set of required services for organizing and conducting a voting process.
2. Support, in accordance to a well-defined operational framework, all ‘actors’ that have a need to interact with the system.
3. Support different ‘types’ of voting processes like polls, plebiscites, inter-organizational elections, general elections etc.
4. Be customisable in respect to the geographical coverage of the voting process, the number of voting precincts, the number of voters, and other specific characteristics of the process like starting date and time, number of candidates etc.
5. Ensure that:
    a. Only eligible persons can vote.
    b. No person can vote more than once.
    c. The vote is secret.
    d. Each vote is counted in the final tally.

e. The voters trust that their vote is counted.

Assuming that the supported voting process is a '*General Election*', which, as compared to polls, internal elections etc, is the broadest and most complicated election process, the functionality that must be exhibited by an internet-based system in order to meet the aforementioned objectives is listed next:

1. **Authorise Actor:** This is the starting point for any interaction with the information system. It provides access to the system functions that a specific actor (organiser, user etc) is authorised to perform.
2. **Define Election Districts**: Define the districts and the corresponding number of candidates that will be represented in the government - according to the number of respective electors.
3. **Define Electors**: All persons above a certain age have the right/obligation to participate in the election process. Persons over a certain age are included in the elector list, unless convicted to attainder or excluded by judicial judgment.
4. **Manage Parties and Candidates:** Notify the system about candidate parties and insert, modify and delete a party's candidates for a specific election district.
5. **Create Ballots**: Each participating party requires a discrete ballot format and a list of its representatives per election district.
6. **Provide Authentication Means**: Create and distribute authentication means to electors in order to allow them to identify themselves during the voting process.
7. **Cast Vote**: The voter is allowed to cast her vote, provided that she has been successfully authenticated. The voter may be supplied with a receipt, confirming that she has voted.
8. **Tally Votes**: Calculation of the number of votes each participating party has received, along with not valid votes. This process cannot be performed before the end of the election.
9. **Verify Result Integrity**: This process takes place in case a voter - or any other interested party - requests to verify that any of the aforementioned election procedures has been conducted properly.

16.2.1.1 Non-Functional (Security) Requirements

The vast majority of security requirements are common to all e-voting systems since they determine the required compliance of the system with the election principles (democracy) and the security and privacy issues dictated by the international legal frameworks. Security requirements are, at a large extent, fulfilled by the voting protocol adopted by the system (refer to 16.3). Specifically, as presented in [44], the security requirements of an internet-based e-*voting system* can be identified in terms of the properties that a voting protocol must exhibit. A short description follows.

Accuracy

Accuracy, also referenced as correctness in [15], demands that the announced tally exactly matches the actual outcome of the election. This means that no one can change anyone else's vote (**inalterability**), all valid votes are included in the final tally (**completeness**) and no invalid vote is included in the final tally (**soundness**).

Democracy

A system is considered to be "democratic" if only eligible voters are allowed to vote (**eligibility**) and if each eligible voter can only cast a single vote (**unreusability**). An additional characteristic is that no one should be allowed to duplicate anyone else's vote.

Privacy

According to this requirement no-one should be able to link a voter's identity to her vote, after the latter has been cast (**unlinkability**). **Computational privacy** is a weak form of privacy ensuring that the relation between ballots and voters will remain secret for an extremely large period of time, assuming that computational power and techniques will continue to evolve in today's pace. **Information-theoretic privacy** is a stronger and, at the same time, harder to obtain form of privacy, ensuring that no ballot can be linked to a specific voter as long as information theory principles remain sound.

Robustness

This requirement guarantees that no reasonably sized coalition of voters or authorities (either benign or malicious) may disrupt the election. This includes **allowing abstention** of registered voters, without causing problems or allowing other entities to cast legitimate votes on their behalf, as well **as preventing misbehaviour** of voters and authorities from invalidating the election outcome by claiming that some other actor of the system failed to properly execute its part. Robustness implies that security should also be provided against external threats and attacks, e.g. denial of service attacks.

Verifiability

Verifiability implies that there are mechanisms for auditing the election in order to ensure that it has been properly conducted. It can be provided in three different forms: a) **Universal or public verifiability** [65] meaning that anyone (voters, authorities or even external auditors) can verify the election outcome after the announcement of the tally, b) **Individual verifiability with open objection to the tally** [59] which is a weaker requirement allowing every voter to verify that her vote has been properly taken into account and file a sound complaint, in case the vote has been miscounted, without revealing its contents and c) **Individual verifiability** which is an even weaker requirement since it allows for individual voter verification but forces voters to reveal their ballots in order to file a complaint.

Uncoercibility

The concept of *receipt freeness*, introduced by Benaloh and Tuinstra [8], implies that no voter should be able to prove to others how he voted (even if he wants to). On the other hand, uncoercibility means that no party should be able to coerce a voter into revealing her vote. Clearly, the notion of receipt freeness is stronger than *uncoercibility*, thus more difficult to achieve [33], especially in online (general) elections.

Fairness

This property ensures that no one can learn the outcome of the election before the announcement of the tally. Therefore acts like influencing the decision of late voters by announcing an estimate, or provide a significant but unequal advantage (being the first to know) to specific people or groups, are prevented.

Verifiable participation

This requirement, often referred as **declarability**, ensures that it is possible to find out whether a particular voter actually has participated in the election by casting a ballot or not. This requirement is necessary in cases where voter participation is compulsory by law (as in some countries, e.g. Australia, Belgium and Greece) or social context (e.g. small or medium scale elections for a distributed organisation board) where abstention is considered a contemptuous behaviour.

16.3 CRYPTOGRAPHY AND E-VOTING PROTOCOLS

Cryptography is naturally used to secure transactions in complex systems where the interests of the participating entities may be in conflict. Not surprisingly, cryptography is one of the most significant tools for securing online voting protocols. While in traditional elections most ideal security goals such as *democracy, privacy, accuracy, fairness* and *verifiability*, are supposedly satisfied, given a well-known set of physical and administrative premises, this same task is quite difficult in online elections. For example, receipt-freeness and verifiability seem to be contradictory: when voting electronically, the very means that allow a voter to verify that her vote was counted properly (e.g. paper receipts, vote encrypting keys, user-selected randomness, etc), may also allow a dishonest third party to force the voter to reveal her vote.

In Section 16.3.1 we highlight several well-known cryptographic models, proposed in the academic literature, for securing remote elections (e.g. Internet voting). In Section 16.3.2 we will discuss some recent cryptographic schemes for securing e-voting at the polling place.

16.3.1 Cryptographic Models for Remote e-voting

Any scheme for remote e-voting must employ some kind of cryptographic transformation to establish secrecy and/or integrity for a set of crucial transactions. Since the first cryptographic protocols for electronic elections [11][20][6], several solutions have been described in academia to deal with the security problems in online voting. We consider how a variety of remote e-voting schemes in the literature apply some of the generic security requirements. We will use the *unlinkability* requirement to attempt a first categorization of the cryptographic schemes, and then we will consider how properties such as *fairness* and *robustness* are established. The notions of *verifiability* and *receipt-freeness* will be examined separately, due to their importance. Depending on the exact phase where the *unlinkability* property is applied on the encrypted votes, the majority of e-voting schemes can be categorized as follows:

- *Unlinkability at the tallying stage*: Unlinkability is achieved at the tallying stage, by taking advantage of the algebraic properties of several public key encryption schemes. In what is known as the *homomorphic model* (e.g. [16][33][46][2][3]), the originally submitted votes are combined and a "sum" of encrypted votes is produced. The encrypted tally can later be decrypted by a set of election authorities. In the *mix-net model* (e.g. [64][39][13][51]), encrypted votes are shuffled (e.g. re-randomization and re-ordering of the list of votes) by a set of mix servers in a verifiable manner.
- *Unlikability at the vote preparation stage*: the voter proves her eligibility to vote and then submits a "blinded" (i.e. randomized) version [11] of her encrypted vote to an election authority for validation. This "blinding" is later removed and the un-blinded, validated vote is anonymously submitted to the election authorities. This model is also known as the *blind signature* model (e.g. [25][55]).

A well known technique to establish *fairness* in any critical system is to share power among several independent entities, hopefully with colluding interests. In the election paradigm, no single authority should violate the privacy of voters or the correctness of the final tally. An extra requirement would be to establish *robustness* against a (reasonably sized) set of entities who may wish to prevent the completion of the election. As a result, a majority of election authorities is usually enough to accomplish a task (e.g. decrypt the final tally). The notion of *threshold cryptography* [21], adapted for several public key encryption schemes, has been a building block for most cryptographic schemes for remote e-voting.

16.3.1.1 The Mix-net model

At a high level, each node in a mix-net *shuffles* and *re-randomizes* the input messages before passing them to the next node in the network. Re-randomization can be either *re-encryption* [57] or *partial decryption* [11] of the input messages in order to increase the entropy of the system. For *verifiability*, each node must also construct a *zero knowledge*[1] proof of correctness that it has accomplished its task without altering, removing, or adding false votes. Correctness can be verified among the mix servers or be universally verifiable [64]. In the universal scenario, each server would construct a *non-interactive* proof of correct transformations, to be later checked during system audit or by any external observer. In their more robust form, mix servers are mutually distrusted and privacy is ensured as long as at least one mix server refuses to divulge its random choices. There have also been proposals for removing misbehaving mix servers, without disrupting the mixing process [53].

In a generic scenario for mixnet e-voting, voters sign[2] and publish their encrypted vote on a public bulletin board: *Unlinkability* is then established at the tallying level, where a set of mix-servers sequentially perform mixing and prove correctness of their computations. By separating the mixing and tallying mechanisms, any interested party could perform the shuffling and provide proofs of correctness [7]. Finally, a sufficiently large set of election authorities cooperate to decrypt the individual encryptions and produce the result of the election.

Mix-nets naturally support write-in ballots, and allow post-election auditing by preserving the complete list of submitted ballots. In comparison with homomorphic elections, the tallying process in mix-net based systems is considerable slower. Late schemes have improved significantly the efficiency of mix-nets (e.g. [1][50][29][36][26]).

16.3.1.2 The Homomorphic model

The idea of combining the encrypted votes in an additive way to construct the final encrypted tally is due to [15][9]. Later, a more practical scheme for large-scale elections was presented in [16], where an exponential version of the ElGamal cryptosystem was used to allow for homomorphic addition. In a generic homomorphic election, each voter signs and publishes an encryption of her vote on a bulletin board. Unlinkability is established during tallying, by “adding up” the encrypted votes without ever decrypting them. Later, a sufficiently large set of

[1] These are prover-verifier interactive protocols, where the prover proves a statement to the verifier and the verifier learns nothing from the prover that he could not learn by himself, apart from the fact that the prover knows the proof [27]. In [3], an interactive proof, where a human who is not computationally capable plays the role of the verifier, was formalized under the notion of *Assisted Human Interactive Proofs*.

[2] Mix-nets could also be adapted to support anonymity in the vote casting procedure, thus preventing the problem of *forced abstention* attacks [39].

multiple authorities cooperate in decrypting the final tally and the results are published on the bulletin board.

Baudron et al [5] proposed an efficient variation of the model in [16] for multiple candidates and races. Damgard et al. [18] proposed a generalization of the Paillier cryptosystem to support very large tallies. An attempt to bring down the costs of such proofs of validity, especially in elections with multiple races and candidates was made in [30].

Homomorphic elections naturally establish universal verifiability and are characterized by a very fast tallying process. Note that each vote must belong to a well-determined set of possible votes such as {+1, -1} for {“yes”, “no”} votes. Moreover, each voter must provide a universally verifiable proof that her vote belongs to the predefined set of votes, or else it would be easy for a malicious voter to manipulate the final tally. Obviously, schemes based on this model seem unsuitable for running elections where votes cannot be combined additively [67].

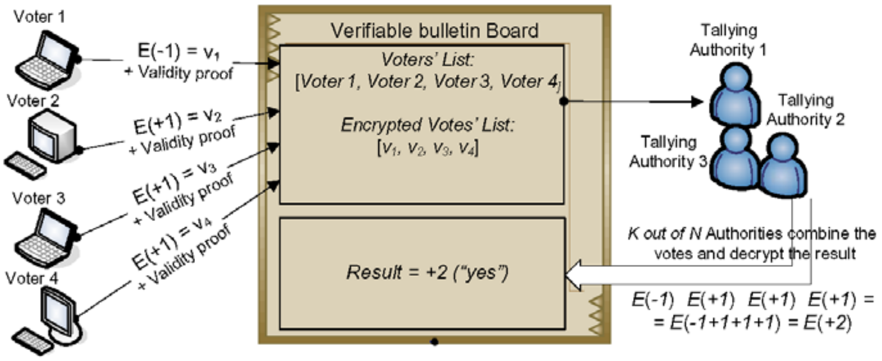

Figure 1. An example election based on the homomorphic model

16.3.1.3 The “Blind Signature” model

Election protocols of this category, introduced in [25], enable voters to get their vote validated by an election authority, while preserving the secrecy of their vote. *Blind signatures* [12] are the electronic equivalent of signing carbon-paper-lined envelopes. In an online voting protocol, a voter encrypts, then blinds the vote, and presents it to an election authority who blindly signs it. Then, the voter removes the blinding factor and gets a validated and encrypted vote that cannot be correlated to the original blinded message. The voter then uses an anonymous channel to submit the validated vote to the election authorities. Later, the voter may even anonymously *object to the tally* [59], if her vote is missing.

Schemes according to this model usually result in a complex election setup phase. Due to the anonymity in the vote casting phase, a series of known internal attacks, such as invalid vote submission by malicious election administrators, have made it difficult to establish universal verifiability. Much trust is placed on the election administrators and the anonymity network, concerning both voter privacy and tally correctness. In recent proposals (e.g. [54][22][38][45]) the power of administration is distributed among multiple authorities to augment the security of such schemes.

Observe that the random factor used in blinding as well as in the vote's encryption could also be used as a receipt in a coercion protocol. In the next section we will discuss receipt-free elections. On the other hand, protocols within this model are simple, easily manageable, computationally efficient and naturally support "write-in" ballots. the model is also easily adapted for elections where the list of voters who actually voted is never published [55], which is a pre-requisite against a specific class of coercion attacks (e.g. the *forced abstention attack* [39], also discussed in the next section).

16.3.1.4 Receipt-Freeness in Remote e-voting

In every cryptographic election, where a vote is to be *encrypted* (or *shuffled*, as we have already seen, by a mix-net), with the help of a public key cryptosystem, the encryption operation needs to be *randomized*[3]: the ciphertext will depend on both the plaintext vote and some random value. Otherwise, trivial chosen-plaintext attacks would be possible, given that usually there is a small set of possible votes in the system. Most generic models, discussed in the previous sections, use some randomness during the vote generation protocol to achieve this level of security. In a generic scheme based on blind signatures for example, the voter chooses randomly a blind factor for her vote to be validated. Similarly, in homomorphic and mix-net generic schemes, voters are required to choose some randomness to encrypt their vote with a randomized encryption scheme. In the mix-net model, mix servers also use randomization for re-encryption or partial decryption purposes. However, it has be shown that the randomness used in voting protocols could also be used to undermine the privacy of a voter: As noted in [33], if a scheme requires the voter to choose her own randomness, then this scheme cannot be receipt-free: the randomness may constitute a receipt[4] in a *coercion* or *vote buying* protocol. The notion of *receipt-freeness* in e-voting was introduced by Benaloh [8], and independently by Niemi and Renvall [52].

Special channels. In cryptographic research for remote e-voting, most proposals for receipt-freeness involve some ad hoc physical assumptions and procedural constraints, for example *untappable channels* (e.g. [55][33][4]), or *voting booths* (e.g. [8][13][51][14][3][63]). An untappable channel may require a physically separated and closed communication medium, e.g. a leased line inaccessible from outsiders. In [33] it was claimed that one-way untappable

[3] Among several randomized encryption schemes with nice algebraic properties are the ElGamal [23], and the Paillier [56] cryptosystems.

[4] In [33] it is shown why the schemes of [8] and [10] are incoercible but no receipt-free. In incoercible (but no receipt-free) voting, a voter may lie about her vote to a coercer, but she may be able to construct a receipt if she wants to sell her vote to a vote-buyer.

channels between voters and authorities constitute a minimal assumption for receipt-free elections.

Schemes in [19][31], assume the existence of a *secondary communication channel* between the voter and election authorities: A vote buyer would have difficulties in tapping both channels (or, doing that for a large population of voters). In [39], an untappable channel during the registration phase (e.g. postal mail) and an *anonymous channel* during vote casting phase, were assumed. These assumptions are weaker than the assumption of [33], in that untappability involves an offline transaction between the voter and the authority, which may happen before the election day, thus being more practical. Furthermore, solving the *forced abstention* threat [39] would require establishing anonymity in the vote casting phase [55][67]. Intuitively[5] however, this could also undermine the public auditability of the final tally [42].

Special proofs of knowledge. Often, a voter needs to verify that a third party (which she does not trust) has performed a correct transformation concerning her vote (e.g. a correct re-encryption in a mix-net [33]). This "receipt" should be *privately verifiable*, i.e. not transferable to a vote buyer. The notion of *designated verifier proofs*[6] [37] has often been used in receipt-free protocols to establish non-transferability of cryptographic assertions (e.g. [33][46][2]).

Any e-voting scheme where the name of the voter who participated in the election is publicly announced is subject to a forced abstention attack [39], where the coercer may simply demand a voter to abstain from voting. Furthermore, in elections where write-in ballots are allowed, the decrypted ballot itself could also constitute a receipt for the vote-buyer. Furthermore, most remote e-voting schemes fail to provide protection, in a practical and affordable way, against an *identity theft* attack (also referred to as *simulation* attack in [39]), where the coercer (or, vote buyer) may collect part or all of the voter's secrets and credentials, and even cast the vote on the voter's behalf. Another difficulty in establishing receipt-freeness in remote e-voting is the *secure platform problem* [62]. In remote e-voting, the PC becomes the voting machine and looses the inherent physical and logical security of precinct systems. An adversary may have access to the client's computed and/or communicated data, either directly (e.g. physical presence[7], man in the middle attacks etc.) or indirectly (e.g. trojans, backdoors, spyware etc). In this way, the attacker may actually control all

[5] At a high level, in any remote e-voting scheme that allows for anonymous vote casting, correctness of the final tally is in question, given the possibility that a (non negligible) set of malicious trustees decide to submit invalid votes.

[6] In a simplistic view, a prover will prove knowledge either of the witness in question, or of the private key of the designated verifier. As a result, the verifier (in our case, the voter) will not be able to transfer knowledge to anyone else.

[7] Unless a physical voting booth is used, the threat of a coercer's watching the voter as she votes cannot be dealt with in remote e-voting. This attack could be mitigated by an election that permits voters to re-vote [39].

electronic[8] communication channels between the voter and the election authorities. In another attack scenario, not directly related to receipt-freeness, the client may become a *zombie* in a *botnet* and be used in Distributed Denial Of Service attacks against other voters or against the election servers.

Tamper-resistant hardware. The work in [47] transformed the assumption of an untappable channel between the voter and election authorities into the weaker assumption of an untappable channel between the voter and a tamper-resistant token. Later, Lee and Kim [46] employed *designated verifier* and *divertible zero-knowledge* proofs to correct a security flaw in [47]. Admittedly, voting with a personal election smartcard would be a costly alternative in the large-scale setting: according to a scenario, all elligible voters would be given, during registration, a voting card (and possibly, some necessary reading peripheral). Furthermore, unless additional access control mechanisms are imposed (e.g. fingerprint identification), the mere use of a smartcard cannot protect against identity theft attacks, where a vote buyer may be in possession of all the credentials and secrets of a vote seller. On the other hand, such devices are expected to be applied to a wide range of applications in the near future, when everybody is expected to store their signing and cryptographic keys in their ID card. It remains to be answered whether e-voting could become an extra application without any extra cost[9].

Admittedly, it seems hard to implement receipt-freeness in remote e-voting without any untappability assumptions. Intuitively, such a scheme would probably tweak the privacy/accuracy tradeoff against accuracy, or would be too complex to implement for large scale elections. This is the main reason why recent cryptographic voting schemes require voters to be physically isolated in a *voting booth* during vote casting. As we will see in Section 16.3.2, the voting booth may indeed guarantee privacy and establish verifiability in a non transferable way.

16.3.1.5 Implementations of the Generic Models

No cryptographic protocol for remote e-voting was ever implemented in a large scale system. On the other hand, several protocols have actually been implemented in small-scale environments. The “blind signature” model has been implemented in several projects, mainly due to its simplicity and flexibility. The first implementations were the Sensus system [17] and the EVOX [32] system. The EVOX system was improved by EVOX Multiple Administrators [22] which in turn was succeeded by the REVS system [38] in an effort to eliminate single entities from disrupting the election. Improved implementations of the REVS system [45] increase the robustness of REVS. This is achieved with a scheme that

[8] In [2] it is said that “an attacker cannot control every communication channel between the voter and the authority”. In this context, a secondary (possibly non-electronic) channel (as in [19][39][31]), is also assumed in an indirect way.

[9] This could also raise the cost of vote-buying. From a security point of view however, such multifunction module could also introduce several new risks [41].

prevents specific denial of service attacks against protocol participants from colluding malicious servers. The REVS system is fully implemented in Java and is publicly available [58].

A series of publications (e.g. [50][51]) marketed by VoteHere.net have led to the implementation of several cryptographic assurances in a real system for polling place e-voting under the mix-net model.

16.3.2 Cryptographic Protocols for Polling-Place e-Voting

A fact in the current polling-place (DRE-based) e-voting infrastructure, is that the integrity of an election is more or less dependent on the correctness of the vendor's software [62][66]. Similarly, most cryptographic schemes for remote e-voting, as the ones described in the previous sections, assume a computationally capable voter, then consider verifiability only at the tallying stage, and more or less ignore[10] the vote generation phase.

Recent proposals [13][51][14][3][63], have established the notion of a *voter-verifiable* election. These are actually hybrid paper/electronic systems for *polling place* voting. However, instead of verifying the voting equipment, an emphasis is given on the voter's verifying the election results in an *end to end* way [60]. In voter verifiable schemes, verifiability comes in three flavors: First, the voter needs to have confidence that her vote is *cast as intended* (also referred to as *casting assurance* [3]). In this context, it is important that the human voter will get casting assurance *without* or with *minimal* external help [51][3]. Verification of correctness for the vote generation stage is not always an all-or-nothing fact. *Cut-and-choose* techniques have been proposed by many recent schemes to establish correctness that can be verified by election officials [14] or by human voters [13][51][3][63] before leaving the polling station. If enough voters perform the audit, then fraud and/or errors will be detected (and even corrected) with a non negligible possibility. Second, the voter needs to have confidence[11] that her vote was *tallied as cast*. During the interaction with the system, a receipt is printed that will permit the voter to verify that the final tally contains her vote. Third, for *public verifiability*, the voter must be sure that the final tally is not tampered with by anyone. A vital issue in all schemes discussed in this section, is that whatever evidence gets the voter from the system will not be transferable to a vote buyer or a coercer. Towards the direction of designing secure systems with relatively low complexity, the use of cryptographic primitives in conjunction with the *voting booth* assumption seems very promising.

[10] In schemes such as [33][47][46] a third party (e.g. the smartcard or an election authority) proves correctness of its random choices during the vote generation phase. However, it is assumed that these proofs are assumingly verified by a software component that the voter trusts beforehand.

[11] For example, in Neff's *MarkPledge* system [51], the voter has confidence that her vote was cast as intended by comparing the confirmation code on the voting machine's monitor with the code that was printed on her receipt. Later, the voter can also check that her encrypted vote, printed on the same receipt, is published on the bulletin board.

Chaum [13] was the first to propose a cryptographic scheme for paper ballots. In [13] the voter is presented with two ballot halves, whose superposition[12] yields the plaintext vote and establishes confidence that the vote represents the correct voter's choice. The voter then destroys one half and keeps the other as a receipt. This scheme was recently evolved into the *Punchscan* system [24]. A variant of the Chaum's original scheme was also proposed in [14] under the name of *Prêt-a-Voter*. All the above schemes use verifiable mix-nets to establish unlinkability at the tallying stage. Another scheme, the *Scratch & Vote* system [3], implements the homomorphic model in paper-based voting. Each ballot contains the candidate names on its left half in random order. On the right half, there are the optical-scan bubbles, a 2D barcode that contains the probabilistic encryptions for each candidate choice, and a scratch surface that hides the random values for each encryption. Each voter can select a second ballot for audit purposes, and casting assurance is established with a cut-and-choose protocol: The voter selects one of the two ballots for auditing, scratches it off and verifies[13] that the ballot was formed correctly. Then, she goes into the booth with the second ballot, fills her choices, and discards the left half of the ballot into a receptacle. Out of the booth, an election official verifies that the scratch surface is intact and publicly discards it. The voter casts (what remained of) the ballot and keeps a copy as a receipt for later verification. All encrypted votes are posted on a bulletin board, and the final tally can be constructed by "adding up" the votes in a publicly verifiable way.

In the *ThreeBallot* voting approach for polling place elections, recently proposed by Rivest [61], end-to-end verifiability is achieved *without* using any cryptography. Each voter in [61] gets a multi-ballot with three identical ballots (except that the ID number on each ballot unique). The voter fills in bubbles in rows corresponding to candidates, in a way that no two ballots will ever reveal the voter's choices. The voter chooses at random one ballot to be kept as a receipt and casts (optically scans) all three ballots. The protocol has been shown to be uncoercible but not receipt-free [61].

16.4 DISCUSSION

Historically, in physical elections, most verifiability checks were delegated to election officials at the precinct, during the voting and counting stages. Accordingly, in electronic communication protocols, when cryptography cannot guarantee by itself all properties of a secure electronic transaction, certain reliance must by placed on the behaviour of a set of third parties. It is not clear whether it

[12] Such techniques use *visual cryptography* [49] to provide an encrypted receipt of the ballot.

[13] In [3] it is suggested that a helper organization assists the voter by providing her some randomness to be used in a challenge response protocol with the voting machine. Or, the organization may help a voter during ballot auditing. Auditing in the Scratch & Vote system requires a barcode scanner and a computer. It is assumed that at least one helper organization will be honest and run correct software. Voters could also do their verifiability checks at home using suitable software.

is realistic to expect that there can be several mutually distrustful, independent parties who can be trusted on crucial security properties in remote e-voting schemes [40][43]. On the other hand, in polling place elections, such parties could be representatives from opposing political parties, or even helper organizations [3]. The goal of a cryptographic protocol is to establish security by trusting a third party as little as possible and on as few security properties as possible. Trust on third parties will never be eliminated, but part of it may be transferred on certain properties of mathematics and cryptography.

Another critical factor is security versus complexity. A secure but complex system is unlikely to be adopted for large scale voting [34]. Schemes for remote e-voting are by default complex protocols: In the absence of a voting booth, high security must be provided mainly by cryptographic means. Any system built upon a highly secure cryptographic remote e-voting protocol would probably suffer considerable usability issues. Recent schemes for paper-based voting (Section 16.3.2), take advantage of the voting booth primitive to protect the voter privacy and add as less cryptography as possible to achieve end-to-end verifiability while maintaining privacy for the encrypted votes. However, all these schemes impose a few additional requirements whose purpose may not be clear to voters [7].

Until today, no cryptographic scheme for remote e-voting or for polling place e-voting has been implemented in a real election of significant scale. Transition to remote Internet voting cannot be a one off step. Towards this transition, it seems natural to take the intermediate step of performing secure e-voting *at the precinct*: Recent cryptographic schemes for paper-based voting seem feasible for large-scale elections and easily implementable in the next future. An open question today is whether such advances will increase or decrease public confidence in the voting process.

References

[1] M. Abe. Universally verifiable mix-net with verification work independent of the number of mix-centers. In: Proceedings of the Advances in Cryptology - EUROCRYPT 98, LNCS Vol. 1403, Springer-Verlag, pp. 437-447, 1998.

[2] A. Acquisti. Receipt-Free Homomorphic Elections and Write-in Ballots. Technical Report 2004/105, CMU-ISRI-04-116, Carnegie Mellon, 2004.

[3] B. Adida and R. L. Rivest. Scratch & Vote - Self-contained Paper-based Cryptographic Voting. In: Workshop on Privacy in the Electronic Society - WPES '06, 2006, to be published.

[4] R. Aditya, B. Lee, C. Boyd and E. Dawson. An Efficient Mixnet-Based Voting Scheme Providing Receipt-Freeness. In: 1st Trustbus 2004, LNCS Vol. 3184, Springer-Verlag, pp. 152-161, 2004.

[5] O. Baudron, P. Fouque, D. Pointcheval, G. Poupard, and J. Stern. Practical Multi-Candidate Election System. In: 20th ACM Symposium on Principles of Distributed Computing, ACM Press, pp. 274–283, 2001.

[6] J. Benaloh. Verifiable Secret Ballot Elections. PhD thesis, Yale, 1987.

[7] J. Benaloh. Simple verifiable elections. In: Workshop on Electronic Voting Technology, Vancouver, BC, Canada, USENIX, August 2006.

[8] J. Benaloh, and D. Tuinstra. Receipt-Free Secret-Ballot Elections. In: 26th Annual ACM Symposium on Theory of Computing, ACM, pp. 544-553, 1994.

[9] J. Benaloh and M. Yung. Distributing the power of government to enhance the power of voters. In: Symposium on Principles of Distributed Computing, ACM Press, pp. 52–62, 1986.

[10] R. Canetti, C. Dwork, M. Naor and R. Ostrovsky. Deniable Encryption. In: Advances in Cryptology – Crypto ’97, LNCS, Vol. 1294, Springer-Verlag, pp. 90–104, 1997.

[11] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. In: Communications of the ACM, Vol. 24(2), pp. 84-88, 1981.

[12] D. Chaum. Blind Signatures for Untraceable Payments. In: Crypto '82, Plenum Press, pp. 199-203, 1982.

[13] D. Chaum. Secret-Ballot Receipts: True Voter-Verifiable Elections. In: IEEE Security and Privacy, Vol 2(1), pp. 38–47, 2004.

[14] D. Chaum, P. Y. A. Ryan, and S. Schneider. A practical voter-verifiable election scheme. In: ESORICS 05, LNCS Vol. 3679, Springer-Verlag, pp. 118–139, 2005.

[15] J. D. Cohen, M. J. Fischer. A Robust And Verifiable Cryptographically Secure Election Scheme. In: 26th Annual Symposium on Foundations of Computer Science, IEEE, pp. 372-382, 1985.

[16] R. Cramer, R. Gennaro, and B.Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In: European Transactions on Telecommunications, Vol. 8 (5), pp. 481-490, 1997.

[17] L. Cranor and R. Cytron. Sensus: A Security-Conscious Electronic Polling System for the Internet. In: Hawaii International Conference on System Sciences, Wailea, Hawaii, 1997.

[18] I. Damgard, M. Jurik and J. Nielsen. A generalization of Paillier's public-key system with applications to electronic voting. Manuscript, 2003. Available at: www.daimi.au.dk/~ivan/GenPaillier_finaljour.ps

[19] I. Damgard and M. J. Jurik. Client/server tradeoffs for online elections. In: PKC ’02, LNCS Vol. 2274, pp. 125-140, 2002.

[20] R. Demillo, N. Lynch, and M. Merritt. Cryptographic protocols. In 14th Annual ACM Symposium on Theory of Computing. ACM, pp. 383-400, 1982.

[21] Y. Desmedt. Threshold Cryptography. In: European Transactions on Telecommunications Vol. 5 (4), pp. 449–457, 1994.

[22] B. W. Durette. Multiple Administrators for Electronic Voting. Bachelor's Thesis, Massachusetts Institute of Technology, May 1999.

[23] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: IEEE Trans. on Information Theory, Vol. 30(4), pp. 469–472.

[24] K. Fisher, R. Carback, and A. Sherman. Punchscan: Introduction and System Definition of a High-Integrity Election System. In IAVoSS Workshop On Trustworthy Elections (WOTE’06), Cambridge UK, June 2006.

[25] A. Fujioka, T., Okamoto, and K., Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In: AUSCRYPT '92, LNCS Vol. 718, Springer-Verlag, pp. 244-251, 1993.

[26] J. Furukawa. Efficient and Verifiable Shuffing and Shuffe-Decryption. In: IEICE Trans. Fundamentals E88-A, 1 (Jan. 2005), pp. 172-189, 2005.

[27] O. Goldreich, S. Micali, and A. Widgerson. Proofs that yield nothing but their validity, or all languages in NP have zero-knowledge proof systems. In: Journal.of the ACM, Vol. 38, pp. 691–729, 1991.

[28] S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In: 14th Annual ACM symposium on Theory of Computing, ACM, pp. 365–377, 1982.

[29] J. Groth: A verifiable secret shuffe of homomorphic encryptions, In: Public Key Cryptography 2003, LNCS Vol. 2567, Springer- Verlag, pp. 145-160, 2003.

[30] J. Groth. Non-interactive zero-knowledge arguments for voting. In: ACNS 2005, LNCS Vol. 3531, pp. 467–482, 2005.

[31] J. Groth, and G. Salomonsen. Strong Privacy Protection in Electronic Voting. BRICS Report Series - RS-04-13-2004, BRICS, 2004.

[32] M. Herschberg. Secure Electronic Voting Using the World Wide Web. Master's Thesis, MIT, June 1997. Available at: http://theory.lcs.mit.edu/~cis/theses/ herschberg-masters.pdf

[33] M. Hirt, and K. Sako. Efficient Receipt-Free Voting Based on Homomorphic Encryption. In: Eurocrypt 2000, LNCS Vol. 1807, Springer, pp 539-556, 2000.

[34] L. J. Hoffman, K. L. Jenkins, and J. Blum. Trust beyond security: an expanded trust model. In: Communications of the ACM, Vol. 49(7), pp. 94-101, 2006.

[35] S. Ikonomopoulos, C. Lambrinoudakis, D. Gritzalis, S. Kokolakis and K. Vassiliou. Functional Requirements for a Secure Electronic Voting System. In: IFIP TC11 17th International Conference on Information Security (SEC2002), Egypt, Cairo, pp. 507-520, 2002.

[36] M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In: USENIX Security Symposium, pp. 339–353, 2002.

[37] M. Jakobsson, K. Sako and R. Impagliazzo. Designated verifier proofs and their applications. In: Advances in Cryptology – Eurocrypt '96. LNCS Vol. 1070, Springer-Verlag, pp. 143–154, 1996.

[38] R. Joaquim, A. Zuquette and P. Ferreira. REVS - A Robust Electronic Voting Systems. In: IADIS'03 International Conference of e- Society, pp. 95–103, 2003.

[39] A. Juels, D. Catalano and M. Jakobsson.. Coercion-Resistant Electronic Elections. In: Cryptology ePrint Archive: Report 2002/165, Available at: http://eprint.iacr.org/

[40] C. Karlof, N. Sastry, and D. Wagner. Cryptographic voting protocols: A systems perspective. In: USENIX Security Symposium, pp. 33–50, 2005.

[41] J. Kelsey, B. Schneier, D. Wagner. Protocol interactions and the chosen protocol attack. In: Security Protocols International Workshop (1997), Springer LNCS, v 1361, pp 91–104.

[42] A. Kiayias and M. Yung. The vector-ballot e-voting approach. In: FC 2004, LNCS Vol. 3110, Springer-Verlag, pp. 72–89, 2004.

[43] P. Kubiak, M. Kutyłowski, and F. Zagórski. Kleptographic attacks on a cascade of mix servers. In: ASIACCS'07, March 20-22, 2007, Singapore, to be published.

[44] C. Lambrinoudakis, V. Tsoumas, M. Karyda, and S. Ikonomopoulos. Secure e-Voting: The Current Landscape. In: Secure Electronic Voting: Trends and Perspectives, Capabilities and Limitations. Kluwer Academic Publishers, 2002.

[45] R. Lebre, R. Joaquim, A. Zïquete, P. Ferreira. Internet Voting: Improving resistance to malicious servers in REVS. In: International Conference on Applied Computing (IADIS'2004), 2004.

[46] B. Lee and K. Kim. Receipt-Free Electronic Voting Scheme with a Tamper-Resistant Randomizer. In: ICISC'02. LNCS,Vol. 2587. Springer-Verlag, pp. 389–406, 2002.

[47] E. Magkos, M. Burmester and V. Chrissikopoulos. Receipt-Freeness in Large-scale Elections without Untappable Channels. In: 1st IFIP Conference on E-Commerce/E-business/E-Government, Kluwer, pp. 683-693, 2001.

[48] L. Mitrou, D. Gritzalis, and S. Katsikas. Revisiting Legal and Regulatory Requirements for Secure e-Voting. In: IFIP TC11 17th International Conference on Information Security (SEC2002), pp. 469-480, Egypt, Cairo, 2002.

[49] M. Naor and A. Shamir. Visual cryptography. In: Advances in Cryptology: EUROCRYPT '94, LNCS vol. 950, Springer, pp. 1–12, 1995.

[50] A. Neff. A verifiable Secret Shuffle and its Application to E-voting. In: 8th ACM Conference on Computer and Communications Security, 2001.

[51] A. Neff. Practical High Certainty Intent Verification for Encrypted Votes, 2004. Available at: http://votehere.com/vhti/documentation/vsv-2.0.3638.pdf.

[52] V. Niemi and A. Renvall. How to prevent buying of votes in computer elections. In ASIACRYPT '94, LNCS Vol. 917, Springer-Verlag, 1994. pp. 164–170.

[53] W. Ogata, K. Kurosawa, K. Sako, and K. Takatani: Fault tolerant anonymous channel. In: 1st International Conference on Information and Communications Security – ICICS. LNCS Vol. 1334, Springer-Verlag, pp. 440-444, 1997.

[54] Ohkubo, M., Miura, F., Abe, M., Fujioka, A., And Okamoto, T. 1999. An Improvement on a Practical Secret Voting Scheme. In: Proceedings of the Information Security Conference – IS'99. LNCS, vol. 1729. Springer-Verlag, pp. 225–234, 1999.

[55] T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In: 5th Security Protocols Workshop '97, LNCS Vol. 1163, Springer-Verlag, pp. 125-132, 1997.

[56] P. Paillier. Public key cryptosystems based on discrete logarithms residues. In: Advances in Cryptology – EuroCrypt '99, LNCS, Vol. 1592, Springer-Verlag, pp. 221-236, 1999.

[57] C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In: EuroCrypt 94, LNCS Vol. 765, Springer, pp. 248–259, 1994.

[58] REVS – Robust Electronic Voting System. Available online: http://www.gsd.inesc-id.pt/~revs

[59] A. Riera, J. Borell, J. Rifà. An uncoercible verifiable electronic voting protocol. In: IFIP-SEC'98 Conference, Vienna-Budapest, pp. 206-215, 1998.

[60] R. L. Rivest. Remarks on The Technologies of Electronic Voting, Harvard University's Kennedy School of Government Digital Voting Symposium. Available at: http://theory.lcs.mit.edu/~rivest/2004-06-01%20Harvard%20KSG%20Symposium%20Evoting%20remarks.txt.

[61] R. L. Rivest. The ThreeBallot voting system, 2006. Avaialable at: http://theory.lcs.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf

[62] A. Rubin. Security Considerations for Remote Electronic Voting Over the Internet. Technical Report, AT&T Labs, 2002. Available at: http://avirubin.com/evoting.security.html.

[63] P. Y. A. Ryan and S. A. Schneider. Prêt a voter with re-encryption mixes. In: Computer Security – ESORICS 2006, LNCS Vol. 4189, Springer, pp. 313-326, 2006.

[64] K. Sako and J. Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In: EUROCRYPT 95, LNCS Vol. 921, Springer, pp. 393–403. 1995.

[65] B. Schoenmakers. A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting. In: Advances in Crytoplogy - CRYPTO '99, LNCS, Vol. 1666, Springer-Verlag, pp. 148-164, 1999.

[66] M. Shamos. Paper v. Electronic Voting Records - An Assessment. 2004. Mimeo, Carnegie Mellon University. Available at http://euro.ecom.cmu.edu/people/faculty/mshamos/paper.htm.

[67] W. D. Smith. New cryptographic voting scheme with best-known theoretical properties. In Workshop on Frontiers in Electronic Elections (FEE 2005), Milan, Italy, September 2005.