Secure national electronic voting system: Pilot implementation

Gheith Abandah* Khalid Darabkh Tawfiq Ammari
Omar Qunsul

Computer Engineering Department
The University of Jordan, Amman 11942, Jordan

abandah@ju.edu.jo, k.darabkeh@ju.edu.jo, tawfiq.ammari@gmail.com,
omar.qunsul@gmail.com

October 30, 2013

Abstract

Electronic voting provides accuracy and efficiency to the electoral processes. World democracies would benefit from a secure e-voting system not only to improve voter participation and trust but also to prevent electoral fraud. However, current e-voting systems are complex and have security weaknesses. In this paper, we describe a secure e-voting system for national and local elections. This system satisfies the important requirements of an e-voting system through state-of-the-art technologies and secure processes. The system relies on homomorphic cryptography, zero-knowledge proofs, biometrics, smartcards, open source software, and secure computers for securely and efficiently implementing the system processes over the various stages of electoral process. Furthermore, we describe the pilot implementations of this system that test the main technologies and processes used. We explain how the used technologies and processes achieve the system requirement. In conclusion, we recommend adopting this system for its security, flexibility, economic, and scalability features.

Keywords: Computer security, electronic voting, homomorphic encryption, zero-knowledge proofs, smart cards, voting kiosks

1 Introduction

There are a wide range of *electronic voting* types that utilize electronic means in the various stages of elections from lists preparation through voting to tallying ballots. These e-voting systems promise many benefits such as accuracy in determining voter intent, saving effort, and speed in counting ballots [1, 2, 3, 4]. The early punch card systems and the optical scan voting systems save efforts in counting

*Corresponding author. Tel.: +962-6-535-5000, Fax: +962-6-530-0813

(a)

(b)

(c)

Figure 1: (a) Covered-face voters pose problems in identity check, (b) Cutting the ID card edge to disallow multiple voting, (c) Dipping the voter's finger in special ink to disallow multiple voting

votes. However, newer e-voting systems such as *direct-recording electronic* (DRE) voting systems and public network DRE voting systems could, in addition to saving effort, provide better accuracy and speed. On the other hand, e-voting systems are complex and are criticized for providing many opportunities for electoral fraud [5, 6, 7].

Many democracies all over the world have serious problems throughout their election processes. These problems include voter lists manipulation, ballots stuffing, voter intimidation, and vote buying. The voting centers are often heavily staffed to administer identity check, voting eligibility, and ballot dispersal. Some staff members unfaithfully enforce the regulations for the benefit of their favorite candidates. Identity check is intricate business in cultures where women or men cover their faces. Moreover, primitive techniques are often used to disallow multiple voting such as cutting the edge of the ID card or dipping the voter's finger in special ink. See the illustrations in Fig. 1. These problems lower trust in the political system and electoral processes and, consequently, adversely affect participation in the political life.

Moreover, the *Arab Spring* revolts are transforming several authoritarian countries into fragile democratic societies [8]. These societies, who have long suffered from suppression, will falsification, election fraud, and mock elections, look forward to new fair election systems that they can trust. The Jordanian Ministry of Political Development suggested starting this project to develop an e-voting system that improves trust in the political system, electoral processes, and participation in the political life. The United Nations Entity for Gender Equality and the Empowerment of Women (UN Women) supports this project particularly to enhance women's political participation.

This paper describes the e-voting system which is developed in this project utilizing state-of-the-art technologies to achieve high accuracy, security, flexibility, scalability, and cost. Moreover, we also describe the pilot implementations that we have built to validate important features used in this system. We believe that this

system can be practically implemented in many countries and would improve trust and participation in the political life. It efficiently achieves the e-voting requirements described below.

In the rest of this introduction, we summarize the main requirements of an e-voting system and review the related work. Section 2 provides a general overview of the proposed system's components and electoral stages. Section 3 describes the technologies that have been used in implementing this system. Section 4 describes the system processes that have been designed to carry out the various electoral stages. Section 5 describes the pilot implementations. Section 6 discusses the system features that satisfy the system requirements and secure it against various attacks. The last section provides the concluding remarks of this paper and outlines the future work.

1.1 E-voting system requirements

Electronic voting systems should fulfill several requirements [9, 2, 10, 11]. In the following paragraphs, we describe the main requirements.

Accuracy The e-voting system does not allow altering or deleting a validated vote and does not count any ineligible vote in the final tally.

Democracy It allows only eligible voters to vote and allows every voter to vote only once.

Privacy It does not disclose the votes of the respective voters and does not allow any voter to prove how she voted. This is a fundamental requirement to avoid voter intimidation and vote selling.

Verifiability It allows anyone to verify that all votes were correctly counted. And in case of electoral disputes, it provides means for rechecking the results.

Security It always satisfies the accuracy, democracy, and privacy requirements and does not allow inside or outside attackers to undermine these requirements. Additionally, it satisfies *reliability*, *availability*, and *data integrity* requirements.

Acceptance It is accepted by voters and candidates who believe that the system is fair and they trust its results. This requirement depends on all above mentioned requirements.

Flexibility It can carry out various types of elections for parliaments, municipalities, student boards, plebiscites, referendums, *etc.* Flexibility provides economic advantage when the same system is used to conduct multiple electoral processes in a certain country or city. A necessary aspect of the flexibility, which is needed for democracy, is the *universality* in allowing any eligible voter to vote irrespective of her native language, special needs, or literacy level. One more important aspect is the *mobility* in allowing the voter to vote in any voting center that is most

convenient to her. There should also be flexibility to allow changing the hardware devices when new or better devices are available.

Cost effectiveness It uses economic software and hardware components. This requirement is particularly important for large-scale elections.

Scalability It efficiently enables carrying out various sizes of elections. The election size can be small (up to few hundred voters, e.g., electing the speaker of a parliament), medium (thousands of voters, e.g., electing the city board), or large (millions of voters, e.g., general parliamentary elections). When the system is used in various election sizes, it achieves flexibility, provides better return on investment, and facilitates productizing it in mass quantities.

1.2 Related work

The literature has many studies that analyze the effectiveness and security of existing e-voting systems. Although voting automation is gaining acceptance, creating comprehensive e-voting systems is hindered by security, verifiability, and acceptance issues [5, 12, 10, 13, 14, 6, 15, 7]. Problems found in existing systems include ineffective voter authentication, lack or insufficient use of cryptography, vulnerabilities of used computers to network attacks, software problems and loop holes, and public distrust.

There have been a number of e-voting systems used in different countries with varying success degrees [16]. Voting over the internet has been adopted in Estonia [17]. The Estonian e-voting experiment shows that, although the technological and logistical requirements might be available, the most important issue is the voter and candidate acceptance. Although Estonia has an IT-literate community (Estonia pays more than any other country on ICT per capita), less than 1% of the constituency voted using the online system. Although voting over the internet is becoming more popular in Norway, it frequently suffers setbacks due to incidents of fraud, privacy concerns, and vulnerability to vote buying and voter coercion [15].

Electronic voting machines are used in India, which is the largest democracy. The Indian system mainly mimics a manual voting ballot box, but replacing the paper with buttons [18]. Although the system works quite well in electronically capturing voter intent, it lacks connectivity and flexibility features. Although the electoral machines themselves are cheap, the cost of moving them under strict security all over India is high.

Several researchers have proposed secure, trustworthy, and scalable e-voting systems [19, 20, 21]. REVS is an e-voting system that allows voters to cast their votes over the internet [10, 22]. The system has excellent robustness and security features. However, internet voting systems are very vulnerable to voter intimidation and vote selling. Paul and Tanenbaum describe an electronic voting approach that incorporates a trustworthy process based on open source software and built-in redundant safeguards that prevent tampering [3]. However, the system depends on the mail system and relies on external measures to check voter identities. VoteBox is a verifiable e-voting system that has the DREs interconnected by local networks

in the *Auditorium* scheme [23]. Each DRE broadcasts its events and records received events for high verifiability. Moreover, the DREs have reduced software stack through using pre-rendered graphics. However, VoteBox does not handle voter registration, eligibility, and list management. ProVotE is an end-to-end e-voting system with a voter verified paper audit trial [4]. It uses formal methods in the design and validation of the voting machines. Similar to [3, 23], ProVotE does not use smartcards or biometrics to check voter identity and improve security.

Baudron *et al.* describe an election system that uses Paillier cryptosystem and zero-knowledge proofs [24]. Paillier cryptosystem has some homomorphic operations that allow tallying the encrypted ballots, thus preserving the voter privacy. However, the used zero-knowledge proofs are inefficient in the limited vote elections.

2 System overview

This section overviews the proposed system by specifying its components and the used electoral process stages. It also states the main assumptions.

2.1 System components

Fig. 2 shows the main components of the proposed voting system. This system consists of enrollment workstations, registrar computer, civil database, voting kiosks, central register, tally decryption computers, and publication server. As shown, these components are interconnected via a network (WAN or the internet). Some of these components such as the enrollment workstations have continuous network connections and are drawn with solid connections. However, other components such as the registrar do not rely on online network connections. Such components are drawn with dashed network connections. These components communicate through interchanging signed files. This interchange can be done through the network or through removable storage media when network connections are unavailable or insecure. The enrollment workstations, registrar computer, and civil database are owned and operated by the specialized *civil information department* (CID). The central register and publication server are housed and operated by a specialized *national information technology center* (NITC) that has secure IT facilities. The following paragraphs briefly describe these components. More detail is available in latter sections.

Enrollment workstations The enrollment workstations are used by the CID staff to register voters, enter and update their data, and issue personal ID cards. These ID cards are smartcards used in the voting process. Each administration workstation is equipped with a fingerprint scanner and a smartcard reader. These workstations are used to capture all needed voter data and update the civil database. It is crucial that this data is entered and updated accurately and smartcards are issued to eligible holders. This accuracy affects all future electoral processes.

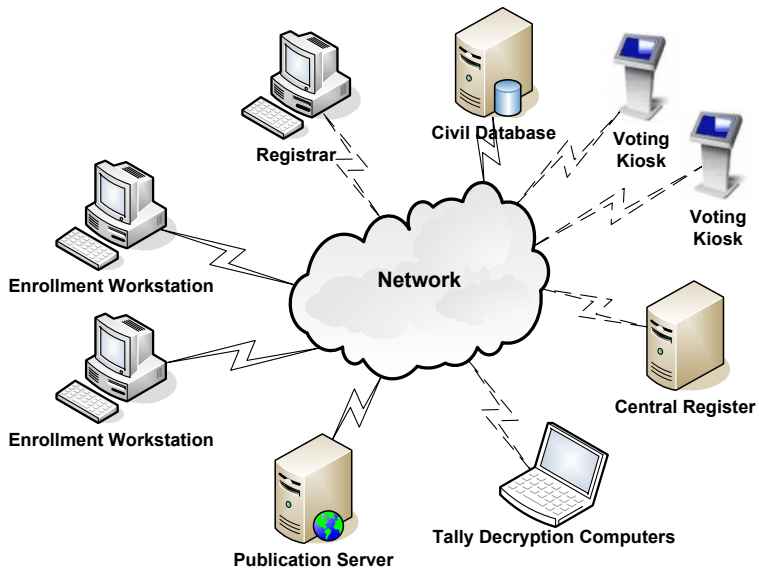

Figure 2: System components. The solid connections to the network are online connections, whereas the dashed connections could be online connections or file transfers through removable storage media.

Registrar computer Every CID branch should have one registrar computer. This is a *secure* computer as described in Section 3. This registrar has a smartcard reader and is used to activate smartcards and extract copies of their data.

Civil database The civil database server is the main CID database server and holds civil records and voter information. The enrollment workstations communicate with the civil server through web services. Web services with secure sockets layer (SSL) provide security features in addition to their excellent access even with the presence of firewalls [25].

Voting kiosks The voting kiosks are located in the voting centers. Each voting kiosk has a touch screen, smartcard reader, fingerprint scanner, and printer. The touch screen provides fast and friendly user interface for casting votes.

Central register This computer is another secure computer where enrollment data, voter eligibility lists, and filled ballots are aggregated and processed. This computer plays a central role in maintaining voter lists, preparing ballots, and verifying and tallying filled ballots. Additionally, the central register exports encrypted voting tallies to the tally decryption computers and exports the voter lists and voting data to the publication server for public verification.

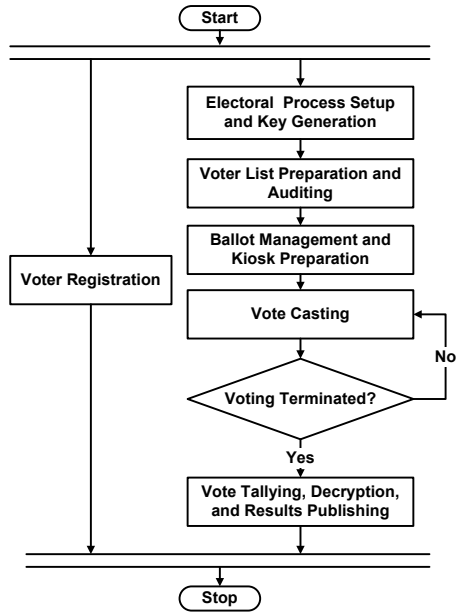

Figure 3: Electoral process flowchart

Tally decryption computers These computers are responsible of decrypting the final voting tallies. They use distributed cryptographic key generation and threshold decryption algorithms as described in Section 3.

Publication server This server is a web server that is used to post electoral information and results. Cryptographic public keys, eligible voter list, filled encrypted ballots, and voting results are posted on this server. Mirror sites for this server can be used to counter denial-of-service attacks and to distribute the load.

2.2 Electoral process stages

The flowchart shown in Fig. 3 summarizes the stages of the electoral process in our proposed system. These stages are summarized below and detailed in Section 4.

Voter registration This registration is conducted using the enrollment workstations and the registrar. It is important to notice that the voter registration is a continuous process that precedes and outlasts a certain electoral process. For example, issuing personal ID cards is a continuous process and a new electoral process is configured every four years in some countries.

Electoral process setup and key generation In this stage, needed setup actions are performed such as cryptographic keys generation and preparation of some system components.

Voter list preparation and auditing As specified by the relevant laws, the eligible voter list should be prepared, published, audited, and finalized according to certain timeline before the election time.

Ballot management and kiosk preparation Just before the election time, the ballots containing the finalized lists of candidates are prepared and securely distributed. Additionally, the voting kiosks are prepared and configured with certified software and ballots.

Vote casting Casting votes occurs in the voting centers using the voting kiosks. As the necessary voter data is available on the smartcard, the voter can vote in the nearest voting center. The kiosk stores in its local storage the encrypted filled ballots.

Vote tallying Once vote casting terminates, the encrypted ballots are securely transferred from the kiosks and aggregated in the central register. The central register verifies the ballots, removes invalid ballots, tallies valid ballots, and exports the encrypted voting tally to the tally decryption computers as described in Section 4.

2.3 Assumptions

For the proper operation of this system, we assume the following assumptions.

- Issuing ID cards is the CID responsibility and is done accurately.
- The techniques described in Section 3 to secure the registrar computer data are followed and no registrar records can be lost. Otherwise, the affected voters need to replace their smartcards.
- The central register is a powerful server with enough computing and storage resources to handle its load.

3 Technologies used

This section introduces the technologies that have been adopted in this system.

3.1 Cryptography

Standard cryptographic techniques are used to improve security [13]. The industry-standard *public key cryptography* (PKC) is used in this system for achieving authentication and confidentiality [26]. Public key cryptography relies on cryptographic key pairs. A key pair for System X K_X consists of private key K_X^- and public key K_X^+. The private key is only known to X; whereas the public key must be made available to the other systems communicating with X. When X encrypts (*signs*) message m using its private key $K_X^-(m)$ and sends it, the receiver validates that X is the source of the encrypted message when it successfully decrypts $K_X^-(m)$ using the public key to retrieve the original message $K_X^+(K_X^-(m)) = m$. When a message

is encrypted using the public key $K_X^+(m)$, the confidentiality is achieved as only the holder of the private key can decrypt it $K_A^-(K_A^+(m)) = m$. Using 2048-bit RSA encryption keys provides security level sufficient for this system at the current state of the art [27].

We also use *digital signatures* for authentication and to protect data integrity. A *signed* message is a message along with its encrypted *digest* $m + K_X^-(H(m))$. The cryptographic hash algorithm SHA-256 is used to compute the message digest $H(m)$. Authenticity and integrity are validated when the digest computed by the receiver from the received message m matches the digest recovered using $K_X^+(K_X^-(H(m)))$.

3.2 Homomorphic cryptography

We use Paillier cryptosystem for its useful *homomorphic properties* in preserving the privacy of votes [28]. Particularly, this system allows finding the sum of encrypted votes by multiplying them. The votes and the tally remain encrypted, thus preserving the privacy of the voters: $K_V^+(m_1 + m_2) = K_V^+(m_1) \times K_V^+(m_2)$.

For flexibility, we support the limited vote election [29]. We allow each voter to select up to O options from C candidates or options. The vote of each voter V_i is encoded as a voting vector $(m_{i,1}, m_{i,2}, ..., m_{i,C})$ where $m_{i,l} = 0$ or $m_{i,l} = 1$ for $l = 1, 2, ..., C$ and $m_{i,l} = 1$ iff the voter V_i chooses the candidate l. The voting vector $(m_{i,1}, m_{i,2}, \ldots, m_{i,C})$ is encrypted to the vector $(c_{i,1}, c_{i,2}, \ldots, c_{i,C})$ and the homomorphic property allows finding the encrypted tally of option l by $\prod_{i=1}^{V} c_{i,l}$; where V is the number of voters.

Paillier cryptosystem has another useful homomorphic property: an encrypted plaintext raised to a constant k will decrypt to the product of the plaintext and the constant. Therefore, negation can be done by raising to $k = -1$. This property can be used to remove some vote $c_{i,l}$ from the tally T_l by $K_V^+(T_l - c_{i,l}) = K_V^+(T_l) \times K_V^+(c_{i,l})^{-1}$.

3.3 Distributed key generation and threshold cryptosystem

To protect voters' privacy, no single party can be trusted to possess the voting decryption key K_V^- that allows this party to uncover how voters have voted. Therefore, we use a threshold version of Paillier cryptosystem [30]. In this system, at least $t+1$ parties should cooperate to decrypt the encrypted tally. Any number of parties t or fewer will not be able to decrypt the tally or any voter's ballot. To achieve this configuration, the key is generated using a distributed key generation algorithm among n parties. It is required that $n > 2t$ and no $t+1$ parties or more would conspire to violate voter privacy. The following paragraphs describe this cryptosystem which is adopted from Nishide and Sakurai work [31]. This description is a brief overview this system. For fuller detail and correctness proofs, kindly see Ref. [31].

Distributed key generation The n parties conduct the following steps to generate an RSA modulus $N = pq$ [32].

1. Every party P_i $(1 \leq i \leq n)$ selects random secrets p_i and q_i of certain properties.

2. The parties adopt a large prime number $P' > \{n(3 \times 2^{k-1})\}^2 > 2N$ where k is the desired key length.

3. The n parties use the BGW protocol proposed in Ref. [33] to compute $N = pq = \sum_{i=1}^{n} p_i \times \sum_{i=1}^{n} q_i \bmod P'$. This step allows computing and publishing the RSA modulus without exposing its factors. Moreover, the parties have at the end of this step a polynomial sharing of Euler totient function $\varphi(N)$ because $\varphi(N) = N + 1 - \sum_{i=1}^{n}(p_i + q_i)$ [31].

4. The parties perform a distributed test to ensure that N is a product of two primes [32]. If the test fails, the parties repeat Steps 1 through 4. This test does not reveal the two primes.

5. Each party P_i selects random integers $\beta_i \in_R [0, KN]$ and $R_i \in_R [0, K^2N]$ where $1/K$ is negligible. The sum $\sum_{i=1}^{n} \Delta R_i$ (where $\Delta = n!$) is shared as polynomial $f_1(x)$ among all parties using Pedersen's verifiable secret sharing [34].

6. By using the BGW protocol, the parties compute and publish $\theta' = \Delta\varphi(N)\,(\sum_{i=1}^{n} \beta_i) + N\,(\sum_{i=1}^{n} \Delta R_i)$. This completes the public key generation $PK = (N, G, \theta')$ and $G = N + 1$. Let $\beta = \sum_{i=1}^{n} \beta_i$ and $\varphi = \varphi(N)$; then $\theta' = \Delta\varphi\beta + Nf_1(x)$. The value $-\Delta\varphi\beta$ is shared as the polynomial $f(x) = Nf_1(x) - \theta'$ which constitutes a distributed secret key $SK = \Delta\varphi\beta$.

Encryption The votes are encrypted into cypher text $c_{i,l} = K_V^+(m_{i,l}) = K_V^+(M) = G^M x^N \bmod N^2$ where x is randomly selected and $x \in Z_N^*$.

Decryption Every party P_i computes and publishes its partial decryption share $c_i = c^{2\Delta f(i)} \bmod N^2$. Then every party can combine $t+1$ partial decryption shares of Subset S of the parties to get $M = L(\prod_{j \in S} c_j^{2\Delta\lambda_{0,j}^S} \bmod N^2) \times \frac{1}{-4\Delta^2\theta'} \bmod N$ where $\lambda_{u,j}^S = \prod_{j' \in S \setminus \{j\}} \frac{u-j'}{j-j'}$ and the function L is defined as $L(u) = \frac{u-1}{N}$. Note that $\prod_{j \in S} c_j^{2\Delta\lambda_{0,j}^S} = c^{4\Delta^2 \sum_{j \in S} f(j)\lambda_{0,j}^S} = c^{4\Delta^2(\Delta\varphi\beta)}$, i.e., this product implements the secret key $\Delta\varphi\beta$.

Throughout the key generation and decryption phases outlined above, the parties interchange zero-knowledge proofs to prove that they have implemented their computations faithfully and selected numbers according to the required properties.

3.4 Zero-knowledge proofs

As the votes are tallied using the encrypted voting vectors, it is necessary to ensure that the encrypted vectors carry valid votes. For example, voter V_i can cheat by submitting the vote $c_{i,l} = K_V^+(1000)$ for his favorite candidate l instead of $c_{i,l} = K_V^+(1)$. Therefore, we require that the voters submit *zero-knowledge proofs* that their voting vectors are valid and adhere to the rules described above without

revealing their votes [35]. Vote verification checks are usually the bottleneck in homomorphic e-voting systems as they usually involve lengthy calculations [29]. For possibly large number of candidates C, the *batched* zero-knowledge proof and verification approach offers excellent efficiency [36]. We adopted a non-interactive version of Protocol 2 described in Ref. [37] which is flexible and efficient, honest-verifier zero-knowledge protocol. This protocol is made non-interactive using the Fiat-Shamir heuristic [38]. In this protocol, the voter proves the following two criteria:

$$\wedge_{l=1}^{C}(K_V^-(c_{i,l}) = 0 \vee K_V^-(c_{i,l}) = 1) \tag{1}$$

$$KN\left[\left(\left(\prod_{l=1}^{C} c_{i,l}\right)/G^O\right)^{1/N}\right] \tag{2}$$

Criterion 1 is a proof that every element in the voting vector is either 0 or 1. Criterion 2 is a proof of knowledge of the N^{th} root and demonstrates that there are exactly O ones in the voting vector. Note that G and N are part of the public key.

3.5 Biometrics

One of the important security safeguards in our system is the use of *Biometrics* for voter authentication [39, 40]. Among the physical biometrics such as fingerprint, iris, retina, and hand geometry, we adopted the fingerprint. The fingerprint biometric offers low cost and high accuracy, user acceptance, stability, and ease of use [41].

During voter registration, the voter's fingerprint is scanned and a reference template of the fingerprint image is computed and stored. The claimed identity of the voter is verified at the voting kiosk through comparing the computed template of the presented fingerprint with the reference template [42].

As some voters have amputated fingers or unreadable fingerprints, the system should allow exceptional authentication through personal identification numbers (PIN).

3.6 Smartcards

Smartcards have successfully been used as smart ID cards in many businesses and countries. In addition to showing the photo and the data of the holder similar to conventional ID cards, smartcards provide many other benefits. The smartcard's non-volatile memory and processing capabilities allow storing personal information securely and performing encryption and authentication functions [43].

Access to smartcards is often regulated through entering the right PIN or presenting the right biometric template. Biometric templates are better than PINs because PINs can be disclosed, but templates are "measures of what you are" [44]. In *template-on-card* (TOC) scheme, the host computer compares the captured biometric template with the reference stored in the smartcard. Positive match unlocks the smartcard's full functionality. However, in the *match-on-card* (MOC) scheme, the captured template is compared by the smartcard's processor [45]. Thus, better

security is achieved as the reference template never leaves the card. There are even new devices that integrate fingerprint scanner in the smartcard reader. MOC is adopted in our system.

The cryptographic capabilities of modern smartcards are utilized in our system. Mainly, smartcards are used to generate unique key pair for every voter, hold the private key securely, and sign the encrypted ballot for authentication.

3.7 Open source software

For this large system, we recommend using free and open source software (FOSS) to reduce cost and improve security [46]. The system should use ready open-source operating systems, components, and applications, whenever is possible. Moreover, the voting software developed for our system should be FOSS [47, 48, 49].

Open source software is usually free and is continually revised by many people. Thus, problems are discovered faster and solutions are made available in a more timely fashion. Code auditing is effective in discovering code hacks and vulnerabilities [5]. Although FOSS might pose an exposure problem initially, it provides higher security and robustness on the long run. However, it is important to make the first release publicly available for a long time before the first election. Making the voting software available to the public also improves the public acceptance of the e-voting system.

A special organization should be established to develop and maintain the open source software of the system and appropriate software development processes should be used [50]. The development team of this organization should be responsible of developing the code, publically publishing its releases, and handling the public's change requests. Another team should be responsible of reviewing changes and certifying the product.

3.8 Secure computers

Our system uses a collection of *secure computers*: registrar, central register, and voting kiosks [51]. These computers run certified software and are not connected with other computers through any wired or wireless networks to minimize external threats. They, however, communicate with other system components through exchanging signed packages on removable storage media such as smartcards and USB Flash memories.

These computers are prepared using the special processes described in the following section. Once the certified software is installed, the secure computer is physically locked to prohibit tampering with its software. Moreover, a seal is placed to detect any violation. Software attestation gives assurance that the running software is authentic [3].

These computers achieve high reliability and availability through storage replication, routine backup, state machine operation, and uninterruptable power supplies (UPS). The data that is maintained by these computers is stored on internal hard disks with digests to allow error detection. The data is also replicated on an external storage device such as USB Flash memory or external hard disk. The external storage also facilitates taking routine backups. Although UPS is available for the

Table 1: Cryptographic keys used

Symbol	Usage	Private key place	Public key user(s)
K_C	Central register authentication	Central register	Publication server, kiosks
K_R	Registration	Registrar	Central register, kiosks
K_D	Eligibility lists	Civil database	Central register
K_V	Vote encryption	Distributed	Central register, kiosks
K_i	Voter signature	Smartcard	Central register
K_K	Kiosk signature	Kiosk	Central register

central register, kiosks may suffer from intentional or accidental power failures. Therefore, the kiosks are state machines and state changes are stored in the hard disk. After recovering from power failure, the kiosk resumes operation at the state it has reached [4].

3.9 Electronic kiosks

Electronic kiosks are widely used in banks and utility companies in the customer and queue management systems. These kiosks usually have rugged case, built-in PC, touch screen, thermal printer, and optional card reader. Such kiosks can be easily adapted to serve as voting kiosks. However, fingerprint scanners need to be attached and some kiosks should be equipped with additional user interface devices for voters with special needs. In case that the elections are held on weekends and to reduce ownership costs, kiosks may be rented from banks and utility companies. This idea can be extended to rent also the reception halls of these companies to conduct vote casting. Well-designed user interface eliminates many voter errors [5, 6].

4 System processes

The system has a collection of processes to fulfill its requirements. The following seven subsections describe these processes.

4.1 Key management

The system uses several asymmetric cryptographic key pairs. Table 1 summarizes these keys through showing the key pair symbol, usage purpose, private key place, and public key place. The management of the first four pairs is described in this subsection. Whereas, K_i is described in Subsection 4.2 and K_K is described in Subsection 4.4.

Fig. 4 shows how the first three key pairs are generated and distributed. The central register's key pair is generated in the central register itself. While, the private key K_C^- never leaves the central register, the public key K_C^+ is transferred through a removable medium (RM) such as USB Flash memory to the publication server. The publication server publishes K_C^+ and makes it available to all system

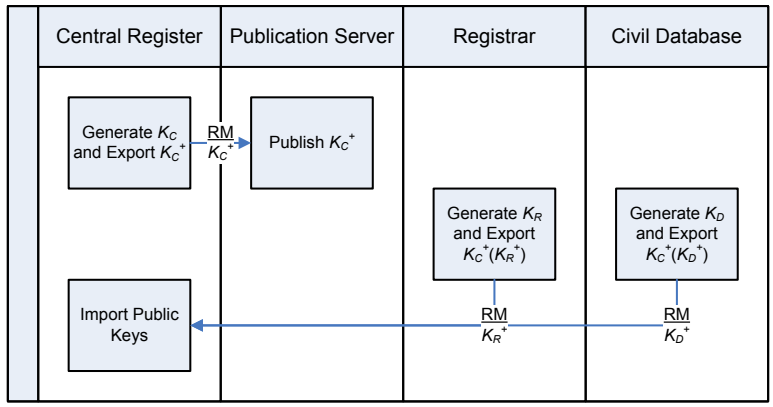

Figure 4: Key generation process. The link labels show the message content below the communication medium (underlined).

components. Similarly, the registrar and civil database key pairs are generated on the respective computers and are transferred to the central register on removable media. All public key distribution is done through proper chains of custody to avoid any fraud. Alternatively, a *certification authority* CA can be used to securely certify the public keys [52].

The voting key K_V is generated using the distributed algorithm described in Section 3.3. A group of n parties is needed to generate the voting key and share secret shares of the decryption key K_V^-. At least $t+1$ parties of this group must collaborate to decrypt the voting tally or to decrypt a voter's vote; where $n > 2t$.

The hard problem is to find such a group that does not contain a subgroup of $t+1$ parties that would conspire to violate the privacy of voters and decrypt their individual votes. We think that a good solution is to have a group of size $n = 7$ and includes technical representatives of the largest three political parties in the country, a head independent judge, and three other independent judges. This group swears that they will not disclose their secret shares and will only use these shares to decrypt the final tally. However, only the head judge and the three political representatives ($t + 1 = 4$) participate in the final tally decryption. Other judges would only participate in case some of the other four parties are absent for any reason.

In this process, each of the seven parties is given a new notebook computer, the certified key generation software is installed on every notebook, and the seven notebooks participate over a private LAN to generate the key. Each party confidentially keeps his/her notebook that holds a share of the secret key. These notebooks are also used in the final tally decryption as described below.

4.2 Voter registration

Fig. 5 summarizes the process of registering a voter. The voter data is captured on one of the enrollment workstations including the voter's fingerprint template

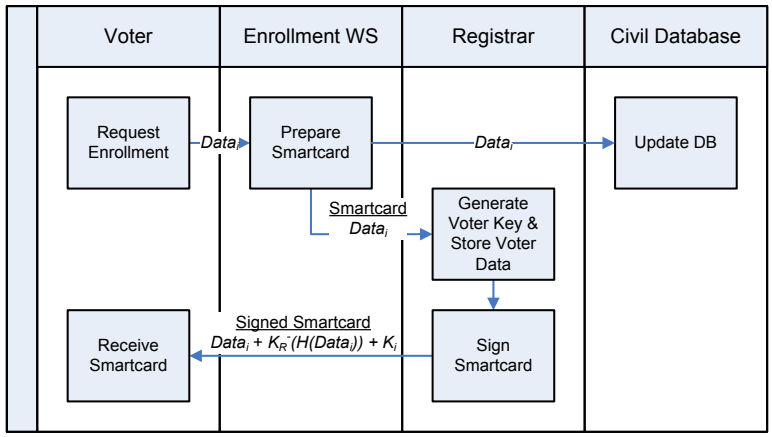

Figure 5: Voter registration process

$Data_i$	Card serial number	Voter ID_i	
	Name	Address	Sex
	Birth date	Precinct ID	Ethnic group
	Preferred language	Special needs	Literacy
	Fingerprint template	Photo	
	Registrar's signature	Private voter key	Voting receipts

Figure 6: Smartcard data

and photo. After performing the necessary checks and approvals, the enrollment workstation updates the civil database and issues an inactive ID smartcard. The smartcard is activated using the registrar computer.

Initially, the registrar instructs the smartcard to generate a unique encryption key for the voter V_i. The private key K_i^- never leaves the smartcard, whereas the public key K_i^+ is kept in the registrar's local storage.

The voter's data is read from the smartcard and is stored in the registrar's local storage. The registrar keeps an append-only local database of registered voters to reduce security threats. The registrar activates the smartcard by adding to the smartcard's data $Data_i$ its signature of the data's digest $K_R^-(H(Data_i))$. This signature is necessary to prohibit attackers from creating or altering smartcards.

Fig. 6 shows the data fields that are kept in the smartcard. The card serial number is a unique read-only card number. The voter ID_i is a unique voter number issued by the CID. The birth date, precinct ID, and ethnic group fields are important in processing voting eligibility. Additionally, these three fields together with the preferred language, special needs, and literacy fields are used by the voting kiosks. Based on these six fields, the kiosk presents to the voter the appropriate ballot in the preferred language and with appropriate user interface for voters of special needs and illiterate voters.

The smartcard holds three additional fields: (i) registrar's signature of the above fields $K_R^-(H(Data_i))$, (ii) hidden voter private key K_i^-, and (iii) IDs of the ballot

types which the voter has filled along with their voting receipts.

4.3 Voter list preparation

Fig. 7 summarizes the process of preparing, auditing, and accrediting the voter list. At the end of the voter registration period, the voters' data is exported from the registrar computers and aggregated in the central register. Every registrar exports one signed package that includes for each voter her data $Data_i$ and her public key K_i^+. The central register aggregates these packages and removes any redundancy, if there is any. Redundancy is expected when multiple smartcards or smartcard updates is issued to the same voter. The central register adopts the last update of a particular voter from its last-issued smartcard. The election committee prepares the list of eligible voters and exports this list from the civil database in a signed package. The central register imports this package and prepares the *initial voter list*. A voter can only appear in the initial list when she has matching records in both the registrar and the civil database packages. The initial voter list is exported from the central database and published on the publication server along with the list of excluded voters and exclusion reasons. In the voter list, the name, precinct ID, sex, age, and ethnic group of every eligible voter is included.

The public audits the initial list, and according to the relevant laws, the process of objections, review, contest, and court decisions is implemented. Then an updated eligibility list is prepared and used in the central register to update the voter list. This update includes deletions, insertions, and changing voters' precincts. Deleting a voter can simply be done by deleting her record from the updated eligibility list. However, inserting a voter can only be done when there is a registrar record for the voter. Nonetheless, this process allows detecting missing registrar packages and importing them. Changing a voter's precinct may create a problem when the updated precinct mismatches the one in the smartcard. The solution for this problem is to include in the packages exported to the kiosks the updated precinct. The kiosk then presents to the voter the ballot of the updated precinct and overrides the precinct ID stored in the smartcard.

At the end of this process, the *final voter list* is exported from the central register and is published on the publication server. This list cannot be changed and is consulted later when the filled ballots are tallied.

4.4 Ballot management

The ballots are designed and maintained on the central register. Each ballot is an Extensible Markup Language (XML) file that includes the following content: ballot ID, precinct or sector, ethnic group, language, one or more questions, two or more candidate answers for each question C, and maximum allowable selections for each question O. The ballot forms are exported and published on the publication server as shown in Fig. 8.

Just before the election time, the central register exports signed kit for preparing the voting kiosks. This kit includes the ballot forms, public keys of the registrars, any voter precinct overrides, and the public voting encryption key. During the preparation process of a kiosk, this kit is verified and imported. Then the kiosk's

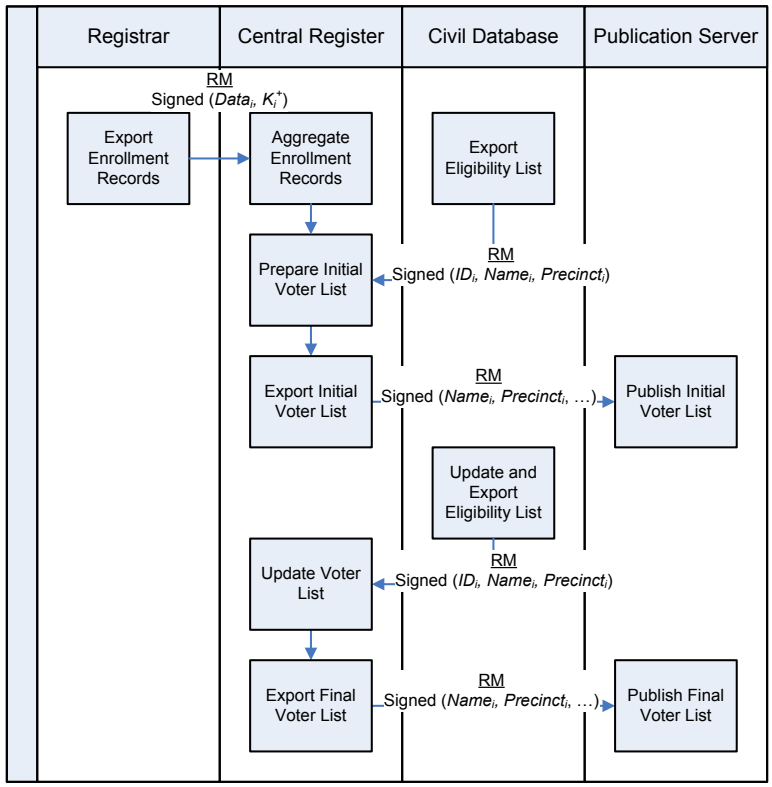

Figure 7: Voter list preparation process

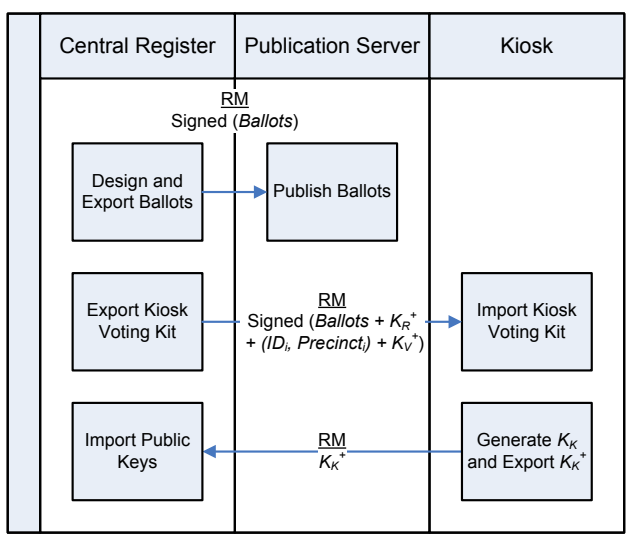

Figure 8: Ballot management process

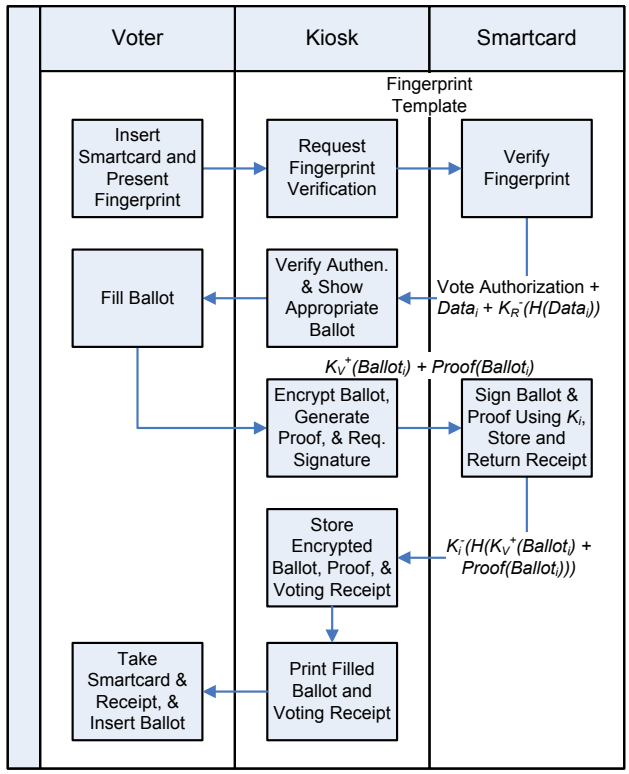

Figure 9: Voting process

encryption key pair K_K is generated and exported to the central database. Similar to the key K_R, the management of this key is through appropriate custody chain.

4.5 Voting

Fig. 9 shows a simplified flowchart for the voting process. The voter starts by inserting her smartcard into the voting kiosk's smartcard reader and presenting her fingerprint to the fingerprint scanner. The kiosk computes the fingerprint template and sends the template to the smartcard for verification. Once the card holder identity is verified, the smartcard is unlocked and the kiosk reads the voter's data $Data_i$ along with its signature. Using the relevant registrar's public key K_R^+, the kiosk verifies the authenticity of the card. Additionally, the kiosk checks the IDs of the voted ballots that are stored on the smartcard to disallow multiple voting by the same voter. After this, the kiosk presents the appropriate ballot form through the suitable user interface according to the relevant $Data_i$ fields.

Once the voter completes filling the ballot, the kiosk generates for every ballot question its voting vector $(m_{i,1}, m_{i,2}, \ldots, m_{i,C})$, the encrypted vector $(c_{i,1}, c_{i,2}, \ldots, c_{i,C})$, and the zero-knowledge proof. The ballot's encrypted vectors $K_V^+(Ballot_i)$ and their ZK proofs $Proof(Ballot_i)$ are sent to the smartcard for signature. The smart-

card signs this ballot by computing the voting receipt using $K_i^-(H(K_V^+(Ballot_i) + Proof(Ballot_i)))$ and returns this receipt to the kiosk. This receipt is also stored in the smartcard along with the ballot ID to disallow multiple voting. For privacy reasons, the kiosk stores, in its local storage, the encrypted ballot vectors, ZK proofs, and voting receipts. It does not keep the plain-text voting vectors.

Finally, the kiosk prints the filled ballot and the voting receipt. The kiosk prints an anonymous hard copy of the filled ballot which must be inserted in a clear ballot box. The printed ballot can be man and machine-readable to allow fast optical scanning in case of a need for recount [53] or to verify the results of the kiosk. This paper trail exposes any kiosk tampering by validating the kiosk's manual count against the homomorphic tally of the kiosk's ballots. As this tally is encrypted, the validation is completed through threshold decryption as described in the following section. The voter keeps a hard copy of the voting receipt to verify that her ballot is ultimately counted [54]. Note that this receipt cannot be used to prove or discover the voter's vote.

4.6 Vote tallying

Fig. 10 shows the vote tallying process. After terminating the voting stage, each kiosk exports a signed package that includes the voter IDs, the encrypted voting vectors, ZK proofs, and the voting receipts. For accuracy purposes, the kiosk does not allow voting after exporting its signed package. The central register verifies authenticity and integrity of the signed packages using relevant public keys K_K^+ and aggregates all ballots.

For each voter ballot record, the central register performs the following checks:

1. The voter's ID_i is in the final voter list.
2. The encrypted ballot carries valid voting vectors. It checks the ZK proof.
3. The encrypted ballot and the ZK proof are authentic and from voter V_i. It checks that $H(K_V^+(Ballot_i) + Proof(Ballot_i)) = K_i^+(Receipt_i)$.
4. The voter does not have multiple voter ballot records.

The central register builds and exports two lists: ineligible votes list and eligible votes list. The ineligible votes list contains any voter ballot record that fails any of the four checks above (with reason). The eligible list contains all other valid records. These lists are published on the publication server so that anyone can verify what votes have reached the final tally. Through publishing voting receipts, voters could verify that their ballots have reached the central register. And through publishing the encrypted eligible ballots, anyone can audit the encrypted tally without violating the voters' privacy.

Then the central register finds the encrypted tally of the eligible encrypted ballots using the homomorphic property of Paillier's cryptosystem described in Section 3.2. It calculates the products of the corresponding fields of the encrypted voting vectors and exports these products to the $t + 1$ tally decryption computers

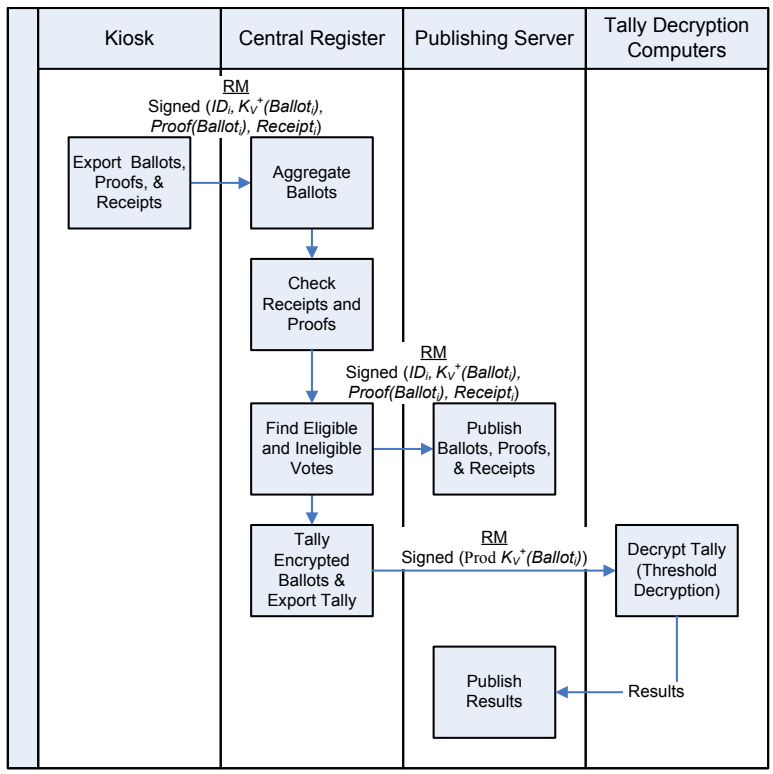

Figure 10: Vote tallying process

described in Section 4.1. These computers perform the threshold decryption algorithm described in Section 3.3 to unveil the final election results and publish them on the publication server.

4.7 Other processes

There are other important processes for preparing the system components. Special processes are needed to install software and to configure the secure components: central register, registrars, and kiosks. Special events are held to prepare these components. Technical representatives of the election committee, NGOs, and political parties should be present to witness these events. In these events, the components are prepared using certified software and signed voting kits before physically looking and sealing them.

5 Pilot implementations

We have implemented pilot projects to test some of the main ideas proposed in this paper. The first pilot system was used in a scientific exhibition in the University of Jordan (National Technology Parade 2008). A total of 784 visitors used this pilot system to vote for the best project presentation in this exhibition. The fingerprint of every voter was scanned and its template was stored on a smartcard. The voter then used the smartcard to cast her vote on the voting kiosk shown in Fig. 11.

This voting kiosk is based on an electronic kiosk used in the customer and queue management systems. This kiosk has a touch screen, thermal printer, and card reader. The thermal printer is used to create a paper trail for auditing purposes. After the voter confirms her selection, she must take the printed ballot and deposit it in the ballot box.

Fig. 12 shows the smartcard and smartcard reader used in this kiosk. They are chosen for their robustness, efficiency, and comparatively low price. This smartcard has enough memory capacity (8 KB) to hold the fingerprint template. However, this smartcards lacks needed features; the more expensive ACOS5-64 smartcard has 64-KB of memory and supports RSA cryptography (up to 4,096 bits) and SHA-256 hashing. Therefore, the ACOS5-64 is suitable for use in the full system described in this paper.

The Microsoft fingerprint reader shown in Fig. 13 was attached to the upper right corner of the kiosk. This particular reader is no longer manufactured or supported but newer scanners that support the standards described in Section 6.1.7 are widely available.

In the first pilot, a commodity PC running Microsoft Windows and equipped with smartcard reader and fingerprint scanner was used as an enrollment workstation. The enrollment and central register programs were developed using Microsoft Visual Studio 2005. Fig. 14 shows the voting management program used on the central register.

This pilot system was successful in testing many usability and efficiency aspects of the voter registration and voting processes described above. Voters were able to use the kiosk and cast their votes in a timely fashion and with minimal directions.

Figure 11: Voting kiosk

Figure 12: Smartcard (ACOS3-32) and smartcard reader (ACR38)

Figure 13: Microsoft fingerprint scanner

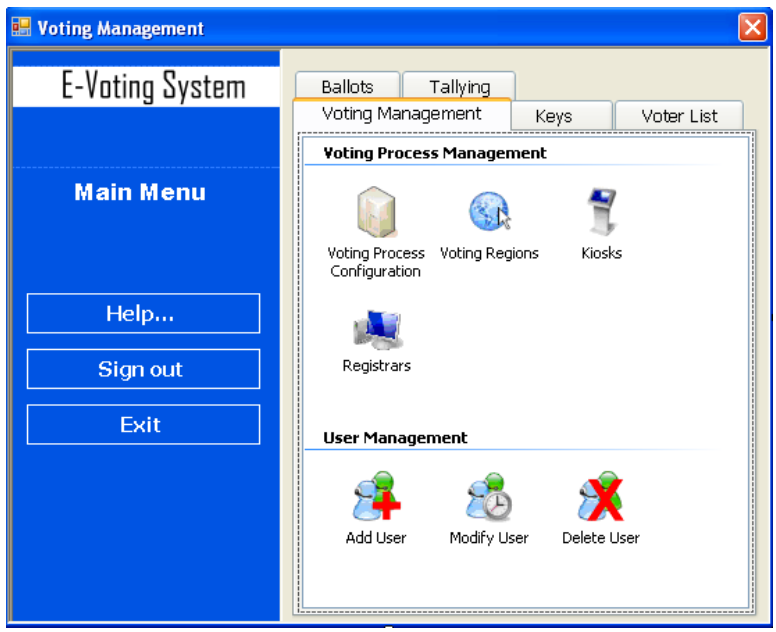

Figure 14: The central register e-voting management program

Table 2: Execution time of key cryptographic functions in milliseconds

Function	$k = 1024$	$k = 2048$
Encrypting a 16-candidate voting vector	170	1,500
Generating a proof for a 16-candidate voting vector	45	36
Verifying vote validity of a 16-candidate voting vector	35	29
Verifying authenticity of the voting receipt	9.1	77
Tallying a 16-candidate voting vector	1.6	3.0

They also gave positive feedback about their voting experience using this system. Generally, they prefer it over the traditional voting techniques used in Jordan.

Later pilots of this system concentrated on developing required cryptographic programs on a Linux operating system using the Java programing language. We have implemented key cryptographic algorithms described in this paper and measured their performance on commodity computers running Ubuntu 10.10. The developed Java programs are based on *The Homomorphic Encryption Project* (http://code.google.com/p/thep/) that implements Paillier cryptosystem in Java, along with its homomorphic operations and key generation. These programs were compiled using GNU Compiler for Java version 4.4.5.

Table 2 summarizes the performance of the main cryptographic functions shown on a computer with Intel Core 2 Quad Q9550 processor running at 2.83 GHz and equipped with 2 GB memory. This table shows the performance for two cryptographic key lengths: 1024 bits and 2048 bits.

These numbers illustrate that the kiosk can encrypt a 16-candidate voting vector and generate its ZK proof in about 1.5 seconds using the large key length. This time is acceptable for the kiosk user. These numbers also illustrate that tallying takes negligible time compared to verifying the vote validity and authenticity. Verifying the vote validity involves checking the ZK proof. Verifying the authenticity involves finding the digest of a 16×4-Kbyte encrypted vector and an 8-Kbyte proof and RSA decryption of the receipt using the voter's public key K_i^+. Whereas tallying a million voting vectors takes around 50 minutes, the verification takes $1,000,000 \times (29 + 77)/3,600,000 = 29$ hours. This time will be even larger with more voters and therefore parallel processing is needed to get the final tally in an acceptable time. For example, an eight-core server can verify 2.2 million votes in a country like Jordan in an eight-hour shift. Larger elections require larger servers.

6 Discussion

This section discusses the system features that satisfy the requirements stated in Section 1.1. Additionally, it analyzes how the system handles the threats that it faces.

6.1 System features

The following paragraphs explain how the system features satisfy each of the nine system requirements.

6.1.1 Accuracy

The system uses secure computers, signed packages, and validated voter list. The votes cast cannot be altered or deleted without detection. They are exported from the kiosks to the central register. Both ends are secure computers and the votes are transferred in signed packages. Any alteration or deletion during this transfer is detected by the central server. Missing kiosk packages are detected as the central register keeps the list of used kiosks and voting receipts are published. In case some ineligible votes are present in these packages, the central register does not count them because it only counts the votes of the voters listed in the final voter list and pass validity and authenticity checks.

6.1.2 Democracy

Eligible voters' authentication is securely carried out using fingerprints and smartcards. Only the real owner of the smartcard ID can cast her vote on the kiosk. This identity check is automated without the need for human involvement. And only the votes of the eligible voters are counted by the central register. Any voter can only vote once because the kiosk stores the ballot type ID on the smartcard preventing the smartcard holder from voting again. Even when this ID is somehow removed, the central register only counts one vote for each voter in the final tally. Multiple votes from the same voter are detected by the central register and published.

6.1.3 Privacy

The system uses encrypted ballots that cannot be decrypted by a single party. The homomorphic property of Paillier cryptosystem is used to tally the ballots without decrypting them. Only the final tally is decrypted through a distributed threshold decryption scheme that involves multiple parties. The voting receipt cannot be decoded to retrieve the vote because this receipt is encrypted digest of the encrypted ballot and the ZK proof.

6.1.4 Verifiability

The system publishes the encrypted ballots, ZK proofs, and voting receipts and uses a paper trail [5, 55, 6, 22]. Anyone can download the eligible ballots from the publishing server and verify the announced tally. Any voter can verify that her vote has reached the final tally through searching for her voting receipts among the published receipts by her ID. The paper trail of ballots deposited in the ballot boxes can be used to verify the electronic results. However, some of these ballots are ineligible for several reasons and are not counted by the central register. For example, a ballot is ineligible when it comes from an ineligible voter. These ballots can be discarded in a physical ballot count when the kiosk gives each ballot a random unique number and when the central register exports the list of ineligible ballots. This physical ballot counting can be sped through using man and machine readable ballots. Paper trails also provide the system with redundancy, thereby making it more robust [6].

6.1.5 Security

The features mentioned above allow the system to securely achieve the accuracy, democracy, and privacy requirements. The use of open source software provides superior secure, reliable, and robust voting software. Software problems and vulnerabilities are detected and solved faster. And the use of secure offline computers for the sensitive system components eliminates all electronic attacks on these components and enhances availability. Moreover, the hardware locks and seals prevent many physical attacks and allow detection of these attacks when committed.

Data exchange among the system components is also secure. Receivers always verify the authenticity and integrity of received data as the exchanged data is digitally signed using hashing and PKC. Furthermore, stored data is secured through replication, digests, and backup as described in the previous sections.

6.1.6 Acceptance

The constituency and candidates are likely to accept the system that achieves the above five requirements. This acceptance is enhanced through providing them with the system's source code and allowing them to check it for any vulnerabilities. Another method to enhance acceptance is using the e-voting system in conducting less critical elections or opinion polls before using it in critical, nation-wide elections. Such elections might be directed at selecting most important historical figures or best sites to visit. Another good bulwark against constituency resistance to the new system is providing them with the feel of the old system, namely, paper ballots. Although adding a paper trail endures additional costs, it is essential to gain acceptance at the early stages of the system adoption and maintain needed verifiability.

6.1.7 Flexibility

A major flexibility feature in our system is a result of using flexible XML descriptions for the ballots [53]. The central register software allows designing any ballot. The voting kiosks import and use the designed ballot forms that are included in the kiosk voting kit. The system supports multiple election types including single vote, limited vote, and approval vote [29]. As the smartcard holds the voter's precinct ID, the kiosk offers the voter the correct ballot according to her precinct. However, when the same system is used in multiple electoral processes, the kiosk cannot rely on one precinct ID. The voter precinct overrides described in Section 4 can specify the ballot that the voter should fill irrespective of the voting precinct. When the national e-voting system is used in electoral processes that include noncitizens, special expatriate smartcard IDs should be distributed.

As all data needed in a vote casting is available on the smartcard and the kiosk, the system allows the voter to vote in any voting center. There is no need to force the voter to appear in a specific voting center that might be far or overcrowded.

The system supports universality through allowing ballots in multiple languages, catering for voters with special needs, and accommodating illiterate people. Illiterates are presented special graphical directions showing candidates' photos or

symbols and supported by audio instructions through headphones.

Hardware can also be changed with minor repercussions. The kiosks are equipped with commodity built-in PCs. The smartcard readers and fingerprint scanners are interfaced through standard interfaces. Thus, minimal changes are needed when changing the interface devices. This is possible through implementing the Public-Key Cryptography Standard (PKCS#11) for interfacing smartcards. Moreover, the ISO standards such as 19794-2:2005 and 7816-11:2004 specify the data format for the interchange of fingerprint minutiae (templates) and personal verification through match-on-card scheme, respectively.

6.1.8 Cost effectiveness

Licensing costs are reduced by using free software and operating systems. The system uses commodity and cheap technologies. For example, it uses fingerprint readers instead of iris scanners for the cost advantage of the former.

6.1.9 Scalability

The system achieves scalability through using a building block (voting kiosk) that can be deployed in various numbers. The major challenge is to provide enough voting kiosks that allow all voters to cast their votes within the limited election period. We estimate that one voting kiosk can serve from 500 to 1,000 voters in a day. This throughput can only be achieved through clear and streamlined kiosk user interface. Depending on the electorate size, the election committee has to provide a suitable number of voting kiosks distributed over voting centers. Aggregating results of a large election in the central register can be sped up through electronically transferring the ballots exported from the kiosks. There is no need to physically deliver the removable storage containing the ballots package exported from the kiosk because the ballots are encrypted and this package is signed. The large computing load of verifying vote validity and authenticity in large countries is manageable and should be handled through parallel processing.

6.2 Threat analysis

The previous discussion explains how the system eliminates or mitigates many security threats. Threats to the system's assets can be physical or through the network, from inside or outside actors, with deliberate motives or accidental, and cause information disclosure, results modification, data loss, or service interruption [3]. The system faces these threats throughout its life cycle. Table 3 summarizes some countermeasures used by the proposed system to face these threats.

The following paragraphs describe how the system fights additional selected security attacks.

If the registrar's private key K_R^- is stolen to make fake smartcards, votes using these cards will not be counted because the data of the fake voters will not reach the civil database and the central register. Hence, these voters will not appear in the final voter list. If this key is stolen to alter smartcards, the central register

Table 3: Security threats and countermeasures throughout the system's life cycle

	Stage	Threats	Countermeasures
1	Product development	Trojan horses, buggy software	Open source software, proper software development methodology
2	Installation and setup	Tampered software or hardware, bad ballot forms	Secure setup processes, certified software, signed setup kits
3	Use by voters	Ineligible voters, multiple votes, denial of service attacks	Biometrics, smartcards, secure computers
4	Transfer and count of votes	Tampered ballots, privacy attacks, incorrect tallying	Signed packages, encryption, ZK proofs, publication of results and ballots

defeats this threat as it uses the original data exported from the official registrar in the tallying process.

If the custody chain fails and a fake K_R^+ key is delivered to the central register, the conspirators could insert fake enrollment records. However, the central register keeps a list of used registrars and imports enrollment records using one public key per registrar. Therefore, if the conspirators manage to insert fake K_R^+ key and enrollment records, the records of the eligible voters enrolled through the official registrar will not appear in the published voter list and the conspiracy will be discovered.

Stealing the kiosk's private key K_K^- or delivering fake K_K^+ is useless without smartcards to sign the ballots. However, stealing this key with vote selling is fought when the central register allows importing only one signed package per kiosk. If the kiosk's signed file is replaced, voters through the official kiosk will discover this when they don't find their voting receipts on the publication server.

7 Conclusions

In this paper, we have proposed a powerful and efficient e-voting system that satisfies the various requirements of e-voting systems. The system relies on specific components to register voters and conduct electoral processes. The enrollment workstations and registrar computers are used to capture voter data and issue signed smartcards. The voter lists are aggregated in the central register that additionally prepares ballots and tallies votes. Voter lists and ballots are transparently published on the publishing server without jeopardizing the voter privacy. The voters use electronic voting kiosks with suitable user interface to authenticate their identities and to fill electronic ballots. These kiosks have smartcard readers, fingerprint scanners, and thermal printers.

The system uses public key cryptography and hashing techniques for privacy, authentication, and validating data integrity. The used smartcards hold the fingerprint templates and allow access when the template of the scanned finger matches the stored template. They also sign the encrypted ballots for authentication and hold voting receipts. These receipts prevent multiple voting and are means for voters to verify that their votes have reached the final tally. The votes are encrypted using homomorphic cryptosystem that allows tallying the ballots without decrypt-

ing them, thus preserving the voters' privacy. The final encrypted tally is decrypted by multiple parties who have secret shares of the decryption key. The selection of these parties minimizes the chance that they would conspire to violate the voters' privacy.

Our system uses open source software to enhance voter and candidate acceptance, reduce costs, and improve security and robustness. The system also uses a collection of secure components to eliminate or reduce security attacks. These computers are not networked, run signed and certified software, and are physically locked and sealed.

The implemented pilot systems permitted testing important technologies and processes used in our system. The results are very promising. A small election process has been conducted efficiently and the user feedback about the voting experience is extremely positive. The kiosk can perform the required vote encryption and ZK proof generation in reasonable time. However, the central register needs parallel processing to verify vote validity and authenticity in large countries.

7.1 Future work

We would like to extend our pilot implementation to a full implementation that covers all aspects of the proposed system including parallel processing of vote validity and authenticity checks. Before adopting this system on a national scale, we would like to test it on a medium electoral process like electing the student union in a large university.

Moreover, we would like to explore integrating internet voting with this system and integrating other applications on the same smartcard, e.g., digital signatures and medical information.

Acknowledgments

This work was supported in part by the United Nations Entity for Gender Equality and the Empowerment of Women (UN Women) and Systems and Electronic Development Company FZCO (SEDCO).

References

[1] N. Ansari, P. Sakarindr, E. Haghani, C. Zhang, A. K. Jain, and Y. Q. Shi, "Evaluating electronic voting systems equipped with voter-verified paper records," *IEEE Security & Privacy*, vol. 6, no. 3, pp. 30–39, 2008.

[2] J. Epstein, "Electronic voting," *Computer*, vol. 40, no. 8, pp. 92–95, 2007.

[3] N. Paul and A. S. Tanenbaum, "Trustworthy voting: from machine to system," *Computer*, vol. 42, no. 5, pp. 23–29, 2009.

[4] A. Villafiorita, K. Weldemariam, and R. Tiella, "Development, formal verification, and evaluation of an e-voting system with VVPAT," *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, pp. 651–661, 2009.

[5] J. Bannet, D. W. Price, A. Rudys, J. Singer, and D. S. Wallach, "Hack-a-vote: Security issues with electronic voting systems," *IEEE Security & Privacy*, vol. 2, no. 1, pp. 32–37, 2004.

[6] D. P. Moynihan, "Building secure elections: E-voting, security, and systems theory," *Public Administration Review*, vol. 64, no. 5, pp. 515–528, 2004.

[7] A. D. Rubin, "Security considerations for remote electronic voting," *Comm. of the ACM*, vol. 45, no. 12, pp. 39–44, 2002.

[8] M. Allansson, J. Baumann, S. Taub, L. Themnér, and P. Wallensteen, "The first year of the Arab Spring," *SIPRI Yearbook*, pp. 45–56, 2012.

[9] T. Antonyan, S. Davtyan, S. Kentros, A. Kiayias, L. Michel, N. Nicolaou, A. Russell, and A. A. Shvartsman, "State-wide elections, optical scan voting systems, and the pursuit of integrity," *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, pp. 597–610, 2009.

[10] R. Joaquim, A. Zúquete, and P. Ferreira, "REVS–a robust electronic voting system," *IADIS Int'l J. of WWW/Internet*, vol. 1, no. 2, pp. 47–63, 2003.

[11] J. Karro and J. Wang, "Towards a practical, secure, and very large scale online election," in *15th Annual Computer Security Applications Conf.*, pp. 161–169, 1999.

[12] A. J. Feldman, J. A. Halderman, and E. W. Felten, "Security analysis of the Diebold AccuVote-TS voting machine," in *2007 USENIX/ACCURATE Electronic Voting Technology Workshop*, 2007.

[13] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach, "Analysis of an electronic voting system," in *IEEE Symp. Security and Privacy*, pp. 27–40, 2004.

[14] T. W. Lauer, "The risk of e-voting," *Electronic J. E-government*, vol. 2, no. 3, pp. 177–186, 2004.

[15] L. Nestas and K. Hole, "Building and maintaining trust in internet voting," *Computer*, vol. 45, no. 5, pp. 74–80, 2012.

[16] G. Schryen and E. Rich, "Security in large-scale internet elections: a retrospective analysis of elections in Estonia, the Netherlands, and Switzerland," *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, pp. 729–744, 2009.

[17] A. H. Trechsel, G. Schwerdt, F. Breuer, M. Alvarez, and T. Hall, "Internet voting in the March 2007 parliamentary elections in Estonia," *European University Institute Paper, Florence*, 2007.

[18] B. Schneier, "Whats wrong with electronic voting machines," tech. rep., Open Democracy, 2004.

[19] P. Y. Ryan, D. Bismark, J. Heather, S. A. Schneider, and Z. Xia, "The prêt à voter verifiable election system," *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, pp. 662–673, 2009.

[20] A. O. Santin, R. G. Costa, and C. A. Maziero, "A three-ballot-based secure electronic voting system," *IEEE Security & Privacy*, vol. 6, no. 3, pp. 14–21, 2008.

[21] J. Tepandi, I. Tsahhirov, and S. Vassiljev, "Wireless PKI security and mobile voting," *Computer*, vol. 43, no. 6, pp. 54–60, 2010.

[22] A. Zúquete, C. Costa, and M. Romão, "An intrusion-tolerant e-voting client system," in *Workshop on Recent Advances in Intrusion-Tolerant Systems*, pp. 23–27, 2007.

[23] D. Sandler, K. Derr, and D. S. Wallach, "Votebox: a tamper-evident, verifiable electronic voting system," in *USENIX Security Symp.*, vol. 4, pp. 349–364, 2008.

[24] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard, "Practical multi-candidate election system," in *20th Annual ACM Symp. Principles of Distributed Computing*, pp. 274–283, 2001.

[25] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Springer, 2010.

[26] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, 2007.

[27] R. Silverman, "Has the RSA algorithm been compromised as a result of Bernsteins paper?," *RSA Laboratories*, vol. 8, 2002.

[28] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptology-EUROCRYPT99*, pp. 223–238, 1999.

[29] J. Groth, "Non-interactive zero-knowledge arguments for voting," in *Applied Cryptography and Network Security*, pp. 467–482, 2005.

[30] P.-A. Fouque, G. Poupard, and J. Stern, "Sharing decryption in the context of voting or lotteries," in *Financial Cryptography*, pp. 90–104, 2001.

[31] T. Nishide and K. Sakurai, "Distributed Paillier cryptosystem without trusted dealer," in *Information Security Applications*, pp. 44–60, 2011.

[32] D. Boneh and M. Franklin, "Efficient generation of shared RSA keys," *J. ACM*, vol. 48, no. 4, pp. 702–722, 2001.

[33] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *20th Annual ACM Symp. Theory of Computing*, pp. 1–10, 1988.

[34] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology-CRYPTO91*, pp. 129–140, 1992.

[35] H. Lipmaa, N. Asokan, and V. Niemi, "Secure Vickrey auctions without threshold trust," in *Financial Cryptography*, pp. 87–101, 2003.

[36] K. Peng, C. Boyd, and E. Dawson, "Batch verification of validity of bids in homomorphic e-auction," *Computer Communications*, vol. 29, no. 15, pp. 2798–2805, 2006.

[37] K. Peng and F. Bao, "Efficient vote validity check in homomorphic electronic voting," in *Information Security and Cryptology–ICISC 2008*, pp. 202–217, 2009.

[38] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in *Advances in Cryptology-Crypto86*, pp. 186–194, 1987.

[39] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[40] S. Liu and M. Silverman, "A practical guide to biometric security technology," *IT Professional*, vol. 3, no. 1, pp. 27–32, 2001.

[41] A. K. Jain, J. Feng, and K. Nandakumar, "Fingerprint matching," *Computer*, vol. 43, no. 2, pp. 36–44, 2010.

[42] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE Security & Privacy*, vol. 1, no. 2, pp. 33–42, 2003.

[43] W. Rankl and W. Effing, *Smart card handbook*. Wiley, 4 ed., 2010.

[44] D. Corcoran, D. Sims, and B. Hillhouse, "Smart cards and biometrics: Your key to PKI," *Linux J*, vol. 59, pp. 68–71, 1999.

[45] G. Bella, S. Bistarelli, and F. Martinelli, "Biometrics to enhance smartcard security," in *Security Protocols*, pp. 324–332, 2005.

[46] D. Riehle, "The economic case for open source foundations," *Computer*, vol. 43, no. 1, pp. 86–90, 2010.

[47] R. Anderson, "Security in open versus closed systems-the dance of Boltzmann, Coase and Moore," in *Conf. Open Source Software Economics, Law and Policies*, 2002.

[48] A. Boulanger, "Open-source versus proprietary software: Is one more reliable and secure than the other?," *IBM Systems J.*, vol. 44, no. 2, pp. 239–248, 2005.

[49] J.-H. Hoepman and B. Jacobs, "Increased security through open source," *Comm. of the ACM*, vol. 50, no. 1, pp. 79–83, 2007.

[50] E. Raymond, "The cathedral and the bazaar," *Knowledge, Technology & Policy*, vol. 12, no. 3, pp. 23–49, 1999.

[51] R. A. Fink, A. T. Sherman, and R. Carback, "TPM meets DRE: reducing the trust base for electronic voting using trusted platform modules," *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, pp. 628–637, 2009.

[52] J. Kurose and W. Keith, *Computer Networking: A Top-Down Approach*. Pearson, 6 ed., 2010.

[53] T. K. Johnson, "An open-secret voting system," *Computer*, vol. 38, no. 3, pp. 98–100, 2005.

[54] A. Yasinsac and M. Bishop, "Of paper trails and voter receipts," in *41st Annual Hawaii Int'l Conf. on System Sciences*, pp. 487–487, 2008.

[55] A. R. Jorba, J. A. O. Ruiz, and P. Brown, "Advanced security to enable trustworthy electronic voting," in *3rd European Conf. on e-Government*, pp. 377–384, 2003.