A secure and audit-able cryptographic e-voting scheme for the Middle East and North Africa

Mona F.M.Mursi, Ghazy M.R.Assassa, Ahmed A. AbdelHafez, Kareem M.AboSamra

Abstract— A new cryptographic electronic voting scheme based on public key cryptography is proposed, to replace the conventional voting methods that are widely used in most developing countries in the Middle East and North Africa (MENA). The proposed e-voting scheme is based on the concept of Prêt à Voter, a paper ballot e-voting scheme. The new e-voting scheme uses paper ballots, due to its familiarity among the public, but with strong cryptographic algorithms with proven security features, to provide enhanced level of ballot secrecy, verifiability and security. The new e-voting scheme eliminates the need for anonymous channels to anonymize the votes in Mixnet based e-voting schemes, yet provides comparable level of security and anonymity with less system complexity. For MENA countries, it is concluded that the replacement of paper-based voting by cryptographic electronic voting to conduct large scale elections is feasible.

Keywords— Cryptography, Electronic Voting, Secret Ballot.

I. Introduction

In the past two decades, electronic voting has got considerable attention as a possible candidate to replace the conventional voting methods. Electronic voting promises to make the electoral process simpler and more efficient for political parties, candidates, election administration, and most importantly for voters. For e-voting to be successful, it should be tailored to meet specific requirements of a particular jurisdiction or country in which they are intended to operate in. The development of an electronic voting model should be based on the requirements of the electoral process as well as the specific needs of voters and other affected parties. Voters in developing countries are familiar with traditional paper ballot voting systems rather than electronic voting machines. Thus the challenge is to create a successful framework upon which an electronic voting model can be effectively developed in those countries. As a case study, Egypt was chosen from MENA countries to develop such framework.

An e-voting framework should be developed to specify in details the functional requirements, which must be tailored to fit the constitutional election principles of the Egyptian voting laws. The framework should also provide the guidelines to design an e-voting scheme to be applicable for direct deployment. The framework should consider the digital divide in Egypt as a main factor that would affect the public acceptance of an e-voting scheme. Therefore among the functional requirements that would be critical is scheme simplicity and familiarity in terms of voters' participation, and their ability to learn and interact with a new e-voting system. This familiarity comes from paper ballots that Egyptians used to cast their votes in their latest presidential elections in 2014 and parliament elections in 2015. This conventional election method asks the voter to register to obtain a paper ballot. The voter then marks the name of his candidate on the paper ballot containing a list of candidates' names using a pen, and then drops his ballot in a transparent yet physically sealed ballot box. The voter then leaves the polling station without having any means of verification that his vote is counted for in the final tally, and thus he is forced to trust the conventional voting system. This trust issue is a major factor in low voter participation in these conventional voting methods.

There exist a set of requirements that an e-voting model should satisfy in order to be successful [1]. Among those requirements is the 'Convenience' requirement, which states that all physical restrictions relative to the voter should be eliminated, and the number of voters having to learn complex techniques in order to vote should decrease. Using paper ballots thus satisfies convenience because voters should be able to cast votes with minimal equipment and skills. Critical requirements related to the security of e-voting systems have to be considered to provide security in terms of vote fraud, vote for others, duplicate votes and voter coercion. An e-voting scheme must protect the privacy of the voters at time of casting the vote and provide ballot secrecy as well.

An e-voting scheme should not have assumptions and requirements that may be difficult to implement on a large scale. Voters should be able to verify that their votes are correctly included in the final tally. Votes should not be able to be modified. Another desirable property relative to the voter is the voter mobility; that the voter need not be restricted to a certain geographical region to cast his vote.

Mona F.M.Mursi is with the Faculty of Engineering, Shoubra, Benha University, 108 Shoubra St, Cairo, Egypt (e-mail: monmursi@yahoo.com).

Ghazy M.R.Assassa is with the Faculty of Engineering, Shoubra, Benha University, 108 Shoubra St, Cairo, Egypt (e-mail: dr.ghazyassassa@feng.bu.edu.eg).

Ahmed A. AbdelHafez is with the Communication Departement, Military Technical College, Kobry Elkobbah, Cairo, Egypt (e-mail: aabdelhafez@gmail.com).

Kareem M.AboSamra is with the Faculty of Engineering, Higher Technological Institute, 3rd area - 7th District - 6th of October, Giza, Egypt (e-mail: karimabosamra@gmail.com).

This paper is organized as follows. Section II reviews the related work present in the literature. In section III, an overview of the new electronic voting scheme is presented. Section IV elaborates our work by developing a set of Unified Modeling Language “UML” use case descriptions for the new e-voting scheme, followed by use case activity diagrams in section V. Section VI details the new scheme and provides an analysis by definition of the scheme, pointing out its potentials to satisfy a wide range of e-voting security properties. Section VII concludes our work, and future work is presented in section VIII.

II. Related Work

There exist many approaches in the literature towards e-voting schemes and systems. Among these approaches is the blind signature approach, which was initiated in [2]. Blind signature is a cryptographic protocol that can be used to authenticate a voter without disclosing the content of his ballot. Blind signatures are the electronic equivalent of signing carbon-paper-lined envelopes. Writing a signature on the envelope leaves a carbon copy of the signature on a slip of paper within the envelope. When the envelope is opened, the slip will show the carbon image of the signature. This approach was introduced for e-voting in [3, 4].

Another approach is the homomorphic encryption approach. The homomorphic property allows the encrypted votes for each candidate to be summed into a single total, without being individually decrypted. This generally applies to “Yes/No” votes. The homomorphic approach was introduced in [5, 6], and was improved in the work of many authors [7]-[11].

Mixnet based approach is one of the main approaches to deploy secret and verifiable electronic elections. Mixnet is a technique to create anonymous channels; a multistage system consisting of cryptography, shuffling and permutations. The function of a Mixnet is to randomize a sequence of mutated messages such that the inputs and outputs of the Mixnet are not link-able. Mixnets in online elections aim at hiding the origin of a ballot so that the link between the identity of the voter and the vote is broken. Messages are mutated either by encrypting & decrypting, or re-encrypting them. The concept of Mixnets was presented by Chaum in [12]. Mixnet approach towards e-voting was introduced and improved in [13]-[16].

General-purpose verifiable Mixnets suffer from some drawbacks as illustrated by Kusters in [17]. In their fully robust form, Mixnets need complex protocols for generating and maintaining shared private keys, as well as for mixing and proving correctness of the shuffles. This affects scalability which makes them suitable for small scale elections with limited number of voters. Another drawback that affects the audit-ability of the elections, is that the amount of data to be verified by observers increase linearly with the number of involved mix nodes, the number of decryptors, and the number of voters, as pointed out by Bernhard in [18] and Chase in [19].

The Prêt à Voter e-voting scheme was proposed by Chaum in [20]. The scheme depends on Mixnets to anonymize the source of an encrypted vote while guaranteeing that the source is valid and that the vote has not been changed. The voting receipts are decrypted and tallied while passing through Mixnets. Since its debut in 2005, the Prêt à Voter scheme had undergone several improvements and developments by many authors in [21]-[26].

PunchScan [27, 28] is a cryptographic voting system that is easy to use by the voter as well as by election officials, while at the same time providing a transparent and reliable process. It incorporates two Prêt à Voter style permutations of the candidate list per ballot, one on each of its two layers.

Scantegrity [29]-[31] is a successor of PunchScan. The system is compatible with US optical scan devices. It provides the voter with a code that is hidden with invisible ink on the ballot, and is revealed with a special pen when the voter marks his candidate, enabling the voter to record the code and use it later to check his vote rather than a receipt.

Scratch & Vote is a variant to Prêt à Voter that use homomorphic tabulation rather than Mixnet. The Scratch & Vote is a cryptographic voting method proposed by Adida & Rivest in [32]. It provides public election audit-ability by proposing the use of scratch strips to allow off-line auditing of ballots. The scratch mechanism also serves to invalidate ballots that have been audited, preventing their use for voting. The method combines a variety of existing cryptographic voting ideas such as homomorphic encryption and the cut-and-choose at the precinct approach.

A non-cryptographic e-voting scheme, which is based on Prêt à Voter but uses scratch strips to mimic the effect of cryptography, is proposed in [33]. Another non-cryptographic approach is ThreeBallot scheme [34], in which the vote is encoded across three ballots, only one of which is kept as the receipt. Another approach is the Farnel based scheme [35], which rests on the observation that verifiability does not require the voter to retain a copy of his own receipt. Accordingly, the Farnel schemes propose mechanisms that allow voters to be given a copy of one or more previously generated receipts. Thus, the annonimization occurs up front, rather than later in the mix/tabulation phase.

In practice, implementing the shuffling of receipts before they are passed out to the voters is difficult without a significant level of trust in procedures and devices. These non-cryptographic schemes do not require an understanding of cryptographic mechanisms for the voter in order to vote. Nonetheless, the assurance arguments are still more subtle than those associated with conventional voting systems. Vulnerabilities in all three of these non-cryptographic schemes have been identified [36, 37], and they do not achieve the same levels of assurance as the more advanced cryptographic schemes.

III. SAC E-VOTING SCHEME

A new electronic voting scheme that is tailored to the chosen case study (EGYPT) is developed based on the concept of Prêt à Voter. The new scheme is named SAC after its key properties; Security and Audit-ability with strong Cryptographic protocols hence the name SAC. Taking into consideration the various requirements for a successful e-voting model, the proposed scheme is built with a secure public key cryptosystem. The scheme will generally operate in three distinct stages; pre-election, vote capture and post-election. An overview of the proposed SAC scheme in these three stages is presented hereafter. Detailed description of the scheme will be discussed in section V.

Stage 1 is the pre-election stage. Pre-election setup requires the election authorities to generate polling stations' digital certificates with the aid of a trusted certificate authority. Each certificate has a public/private key pair. Polling stations are provided each with a public key (encryption key) from the set of polling stations' digital certificates. The polling station private key (decryption key) is hidden by dividing it into shares using a verifiable threshold share technique [38, 39].

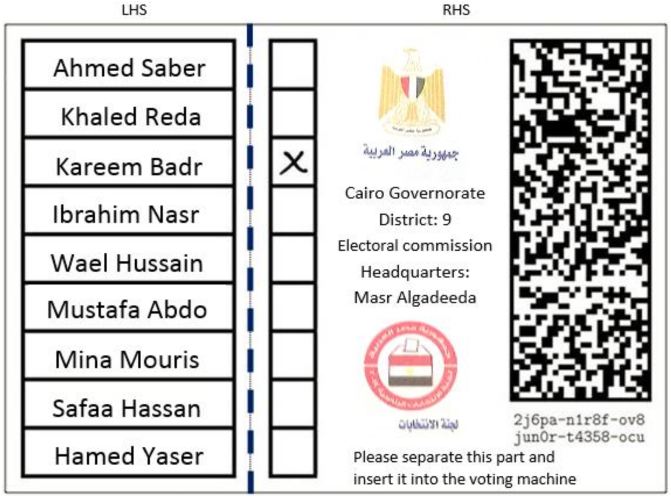
Pre-election setup also requires the generation of paper ballots with randomized candidate list. Election authorities will generate a set of election authorities' digital certificates with the aid of a trusted certificate authority. Since there is a handful of possible candidate names' shifts, for a small candidate list, a permutation technique is used to provide further randomization. Ballots will show different order of candidate names according to the permutation key which is unique for every ballot. The permutation keys are hidden by encryption using the election authorities' public keys. The election authorities' private keys (decryption keys) are also hidden by using a verifiable threshold share technique.

For added security, the election authorities' encryption keys may be threshold shared among several parties. This prohibits a single party from generating ballots without the official approval of other parties sharing the key.

A third set of digital certificates with public/private key pairs are generated to be used in digital signature of the votes. The private keys are stored on smart cards secured with a personal identification number "PIN". These cards are distributed to polling stations just before the election period.

The paper ballot needs to be generated with a specific list of candidates for each polling station. The paper ballot is divided into two parts; left hand side (LHS) and right hand side (RHS). The two parts are printed on a single piece of paper with a perforated line separating the two parts. The LHS contains the names of candidates in permuted order. The RHS contains check boxes corresponding to each candidate and a barcode that hides the encrypted permutation key. Some other information is also printed on the RHS such as the governorate and the official stamp of the election authorities.

A sample of the SAC ballot's layout translated from the native Arabic language is shown in Fig. 1. The voter does not need to participate in any of the pre-election setup processes but he should have a general understanding of the whole voting process.



LHS

RHS

Ahmed Saber □
Khaled Reda □
Kareem Badr ☒
Ibrahim Nasr □
Wael Hussain □
Mustafa Abdo □
Mina Mouris □
Safaa Hassan □
Hamed Yaser □

جمهورية مصر العربية

Cairo Governorate
District: 9
Electoral commission
Headquarters:
Masr Algadeeda

لجنة الانتخابات

2j6pa-n1r8f-ov8
jun0r-t4358-ocu

Please separate this part and
insert it into the voting machine

Fig. 1: Sample of a correctly completed ballot form

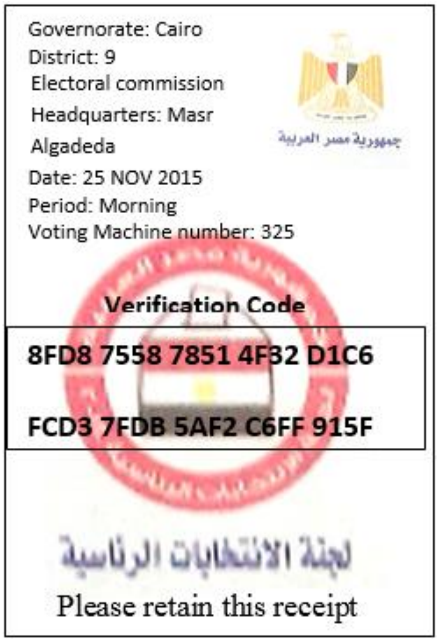
Stage 2 is the vote capture "cast vote" stage. The voter obtains a ballot after registration at the polling station. The voter privately marks his vote on the RHS of the ballot inside a voting booth. Then he splits the ballot into two parts following the perforated line, and feeds the RHS to the voting machine present inside the voting booth. The voting machine scans the RHS and records the data electronically.

The voting machine then marks the RHS of the ballot as voted by printing an election stamp on the ballot's RHS. The election stamp will contain some data related to the election such as date, voting period and voting machine number. This process prevents fraudulent future rescan of the ballot. A unique ballot identifier will be generated that will serve as a verification token in the post-election stage.

The voting machine then performs multiple encryptions to seal the electronic record of the ballot and verification token in a digital envelope. The digital envelope is then digitally signed. The next step is to send the sealed envelope to be stored in a remote database immediately after the vote is cast, or in batches after a certain predetermined time. The digitally signed digital envelope authenticates the source (polling station) while securing the valid vote and verification token from being altered in any way till it is officially opened by the election authorities in stage 3.

The RHS of the ballot is then mechanically dropped by the voting machine in a transparent yet physically sealed ballot box inside the voting machine. This process is necessary for storing physical evidence of the votes, and possibly for the need of an on sight auditing at the polling station. The voting machine then prints the election stamp and the verification token on paper and presents it to the voter as a voting receipt.

A sample of the voting receipt translated from the native Arabic language is shown in Fig. 2. The left part (LHS) of the ballot serves as the voter's ticket out. The voter must submit the LHS of the ballot to an election officer for shredding and then leaves the polling place with his voting receipt.



Governorate: Cairo
District: 9
Electoral commission
Headquarters: Masr Algadeda
Date: 25 NOV 2015
Period: Morning
Voting Machine number: 325

جمهورية مصر العربية

Verification Code

8FD8 7558 7851 4F32 D1C6

FCD3 7FDB 5AF2 C6FF 915F

لجنة الانتخابات الرئاسية

Please retain this receipt

Fig. 2: Sample of a voting receipt

Stage 3 is the post-election stage. Post-election includes tallying of votes and announcing the results. The Talliers are a subset of the election authorities that are entrusted to tally the votes. The Talliers will verify the digital signature of the stored digital envelopes. Then the Talliers open the digital envelopes under supervision from auditors and international monitors. The Talliers will then post the verification tokens to a bulletin board. The bulletin board is public and visible to all. The voters will be able to verify their receipts to match the posted data. The Talliers will not post any other data to preserve ballot secrecy. This will ensure the voters that their votes were successfully received and officially processed towards the final tally.

Talliers will then retrieve the original candidate list after performing some cryptographic procedures. Electronic tallying of votes follow by matching the position of the voter's mark on the check box representing the vote with the corresponding candidate's name, and incrementing the votes count in favor of that candidate. The results are announced on bulletin board or through other public announcement means.

The verification tokens are used for verification purposes only and are not included in the tally. This breaks the link between the encrypted record of the vote and the reconstructed one. This in turn eliminates the need for further shuffling and permutation of votes with complex proof of shuffles, that Mixnet based e-voting schemes need to prove non-fraudulent mixing. It also eliminates the technical difficulties imposed by a failing mix server, therefore the new SAC scheme gains potential for a large scale deployment.

In this paper, the *SAC* e-voting scheme is presented in the context of various assumptions regarding an e-voting system and its processes that provide other aspects of an overall e-voting system. The scheme focuses on obtaining the votes from the voters, and processing those votes towards an election result. It is important to recognize the assumptions which support the claims of a trustworthy e-voting system namely; electoral roll, chain of custody, privacy of voting booth and bulletin board. These assumptions are presented hereafter.

a) *Electoral roll*. It is assumed that the electoral roll is accurately maintained and that voters are suitably authenticated and given only one ballot after registration.
b) *Chain of custody*. The integrity and secrecy of the ballot forms and polling station keys are assumed to be ensured from the time of their creation to the time of use.
c) *Privacy of voting booth*. A reasonable assumption is that the voters are able to cast their vote in private, without the possibility of being observed, and that vote casting takes place in a controlled environment.
d) *Bulletin board*. The scheme requires information to be posted publicly, so that voters and public auditors can access the information needed to carry out their verification checks. The bulletin board is assumed to provide a way of publishing that information. It is important to assure that information is stored reliably on the bulletin board in a tamper-proof way [40, 41].

IV. UML Use Case Description

In order to further clarify the *SAC* e-voting scheme, a set of Unified Modeling Language "UML" use case descriptions are developed. These use case descriptions will be the building blocks of an e-voting system suitable for conducting general large scale elections. The three main stages of the SAC scheme are developed through the following use cases: Pre-election Administration, Cast Vote & Tally Votes. Samples of the developed UML diagrams are presented in the next section of this paper.

The three main stages of the proposed scheme will be described in details in Tables I-IV. The developed use case description tables include primary actors, secondary actors, goal of the use case, trigger of events, relationships with other use cases, inputs from previous use cases, pre-conditions, post conditions on success and on failure and outputs to other subsequent use cases. It will also describe normal flow of events and alternate/exceptional flow of events when needed. Each of the actors performs one step or more in the basic flow of events.

The use case description will also include functional test cases ”FT” for future expansion of the *SAC* e-voting scheme, to accommodate all aspects of a full electronic voting experience. All of the use case activity diagrams, description tables and its corresponding function tests can be found in [42].

TABLE I
PRE-ELECTION

Use Case Description	
System: SAC	**UC ID: 1**
Use Case name: Pre-election Administration	**Priority :** High
Primary actors: Election authority, Certificate authority	**Secondary actors:** Civil group representatives, International monitors, Auditors
Goals ▪ Generate encryption/decryption keys ▪ Generate key shares ▪ Generate ballots	
Trigger: Pre-election preparations	
Relationships ▪ Includes use case: Generate polling station encryption and decryption key (UC ID: 2) ▪ Includes use case: Generate election authority encryption and decryption key (UC ID: 3) ▪ Includes use case: Generate paper ballots (UC ID: 4) ▪ Includes use case: Threshold share polling station decryption key (UC ID: 5) ▪ Includes use case: Threshold share election authority decryption key (UC ID: 6)	
Inputs ▪ List of eligible voters for each polling station ▪ List of Candidates for each polling station	
Pre-conditions: None	
Normal (Basic) flow of events	
Certificate authority **1.** Generate polling station's encryption key (public key) and decryption key (private key) (UC ID: 2) [FT 1.1] **2.** Generate election authority's encryption key (public key) and decryption key (private key) (UC ID: 3) [FT 1.1] **3.** Generate election authority digital signature key (private key) [FT 1.1] **Election authority** **4.** Generate Paper Ballots (UC ID: 4) **5.** Threshold share polling station's decryption key (UC ID: 5) [FT 1.2] **6.** Threshold share election authority's decryption key (UC ID: 6) [FT 1.2] **7.** Destroy election authority's encryption key	
Alternate and *Exceptional* flows: None	
Post-conditions on success: Ballots and encryption keys are generated, decryption keys are threshold shared **Post-conditions on *failure*:** Ballots are not generated, keys are not generated	
Outputs: ▪ Paper ballots and encryption keys are generated and ready for distribution to polling stations ▪ Decryption key shares are ready to be distributed between primary and secondary actors	
Test Cases: FT 1.1: Verify the generation process of keys. FT 1.2: Verify the Threshold share process.	

TABLE II
GENERATE PAPER BALLOTS

Use Case Description	
System: SAC	**UC ID: 4**
Use Case name: Generate paper ballots	**Priority** High
Primary actors: Election authority	**Secondary actors:** None
Goal: Prepare paper ballots	
Trigger: Pre-election preparations	
Relationships ▪ Included in: Pre-election administration use case (UC ID: 1)	
Inputs ▪ List of candidates for each polling station ▪ Election authority's encryption key	
Pre-conditions ▪ Generate polling station encryption and decryption key use case is completed successfully (UC ID: 2) ▪ Generate election authority encryption and decryption key use case is completed successfully (UC ID: 3)	
Normal (Basic) flow of events	
Election authority **1.** Prepare an unordered list of candidates L **2.** Perform N permutations on the candidate list **3.**Encrypt the permutation key N with the election authority's encryption key **4.** Represent the cipher as a two dimensional bar code **5.** Print the permuted list of candidates on the LHS of the ballot and the two dimensional bar code on the RHS of the ballot. [FT 4.1]	
Alternate and *Exceptional* flows: NONE	
Constraints: 1. L is greater than or equal 2 **2.** N is co-prime to L	
Post-conditions on success: Total number of paper ballots for a specific polling station is increased by 1 **Post-conditions on failure:** Total number of paper ballots for a specific polling station remains unchanged	
Output: Paper ballots are generated and ready for distribution to polling stations	
Test Cases: FT 4.1: Examine samples of the generated paper ballots to verify their authenticity and proper construction	

TABLE III
CAST VOTE

Use Case Description	
System: ***SAC***	**UC ID: 7**
Use Case name: Cast Vote	**Priority:** High
Primary actors: Voter	**Secondary actors:** Election authority, Election officer, Civil group representatives, International monitors, Auditors
Goal: Allow a Voter to cast his vote	
Trigger: Start voting period	
Relationships ▪ Include use case: Generate verification token (UC ID: 8) ▪ Include use case: Generate symmetric key (UC ID: 9) ▪ Include use case: Generate digital envelope (UC ID: 10)	
Inputs: ▪ Paper ballot ▪ Polling station's encryption key	
Pre-conditions: ▪ Pre-election administration use case is completed successfully (UC ID: 1)	
Normal (Basic) flow of events	
Voter **1.** Obtain a paper ballot **2.** Mark the vote on the RHS of the ballot **3.** Split the ballot to two parts with the LHS containing the permuted candidate list and RHS containing his marked vote **4.** Feed the RHS of the ballot into the voting machine **Voting Machine** **5.** Scan the RHS of the ballot and record the data that represent the vote and barcode **6.** Apply a hash function to the recorded data and present the digest as the verification token (UC ID: 8) **7.** Generate a symmetric key (UC ID: 9) **8.** Encrypt the vote data with the symmetric key **9.** Encrypt the symmetric key and verification token with polling station's encryption key (UC ID: 10) **10.** Sign the encrypted data, encrypted symmetric key and encrypted verification token with the election authority digital signature key **Voter** **11.** Obtain the voting receipt from the voting machine **12.** Submit the LHS of the ballot to an election officer for shredding then walk away	
Alternate and *Exceptional* flows: **4.1** Voter feeds the RHS of the ballot in upside down a. His vote is discarded	
Parallel Action*:* **7.1** Mark the RHS of the ballot as voted with an election stamp containing the verification token **8.1** Mechanically drop the marked ballot in a physically sealed transparent ballot box **9.** Print the election stamp containing the verification token as a voting receipt. [FT 7.1] **11.1** Send the envelopes in batches to be stored securely in a remote database **12.1** Receive flag of correct recording	
Post-conditions on success: The total **number** of votes for the polling station is incremented by 1 **Post-conditions on *failure*:** The total number of votes for the polling station remains unchanged	
Outputs: ▪ Paper ballots are marked with the voters' choices and dropped in a physically sealed ballot box. ▪ Encrypted digital representation of the vote and verification token are sent outside the polling station in digital envelopes for secure storage	
Test Cases: FT 7.1: Verify the authenticity of the voting receipt	

TABLE IV
TALLY VOTES

Use Case Description	
System: SAC	**UC ID: 11**
Use Case name: Tally Votes	**Priority :** High
Primary actors: Election Authority, Talliers/System	**Secondary actors:** Civil group representatives, International monitors, Auditors, Voters
Goal: Tabulate votes	
Trigger: Signal the end of voting period	
Relationships ▪ Include use case: Reconstruct polling station decryption key (UC ID: 12) ▪ Include use case: Reconstruct election authority decryption key (UC ID: 13) ▪ Include use case: Decrypt ballots (UC ID: 14)	
Inputs: ▪ Polling station's decryption key shares (Output of UC ID: 5) ▪ Election authority's decryption key shares (Output of UC ID: 6) ▪ Digital envelopes (Output of UC ID: 7)	
Pre-conditions: ▪ Pre-election Administration use case is completed successfully (UC ID: 1) ▪ Cast Vote use case is completed successfully (UC ID: 7)	
Normal (Basic) flow of events	
Election authority	

1. Verify the digital signature of the digital envelopes to validate its source and its authenticity
2. Reconstruct the polling station's decryption key with the aid of certificate authority (UC ID: 12) [FT 11.1]

Talliers/System
3. Open the digital envelopes
4. Decrypt the verification token with the polling station's decryption key
4. Post verification tokens to a bulletin board

Voter
5. Voter checks the posted token against his voting receipt

Talliers/System
6. Decrypt the symmetric key with polling station's decryption key
7. Decrypt the encrypted data that represent the ballot with symmetric key
8. Retrieve the digital representation of the ballot

Election authority
9. Reconstruct the election authority's decryption key with the aid of certificate authority (UC ID: 13) [FT 11.1]
10. Election authorities submit their decryption key to Talliers

Talliers/System
11. Talliers begin decryption of permutation key and retrieve candidate list
12. Talliers tabulate the results [FT 11.2]

Election authority
13. Announce the results

Alternate and *Exceptional* flows:
3.1 Fail to decrypt the verification token
a. Source of the envelope is not authentic
b. Discard the envelope
5.1 No match or absent
a. Voter officially complains
b. Auditors check integrity of voter's polling station by auditing the sealed ballot boxes
11.1 Decryption fails
a. Unofficial ballot detected
b. Auditors reject the ballot

Post-conditions on success: The total number of valid votes to be tabulated is incremented by 1
Post-conditions on *failure*: The total number of valid votes to be tabulated remains unchanged

Output: A list of the number of votes for each candidate in each polling station

Test Cases:
FT 11.1: Ensure proper reconstruction of keys and sufficient key shares above threshold are present
FT 11.2: Auditors and International monitors may test the tabulation process

V. UML Activity Diagrams

The UML use case description tables presented earlier describe the main features of the SAC e-voting scheme. Fig. 3 is a simple UML activity diagram of the *SAC* e-voting scheme. The figure shows the main procedures that build the SAC e-voting scheme and the actors' roles in those procedures. Each main stage of the SAC e-voting scheme and its associated activities are represented by activity diagrams showing the basic flow of events and some alternate flow of events. Some of those activity diagrams are presented later in this section.

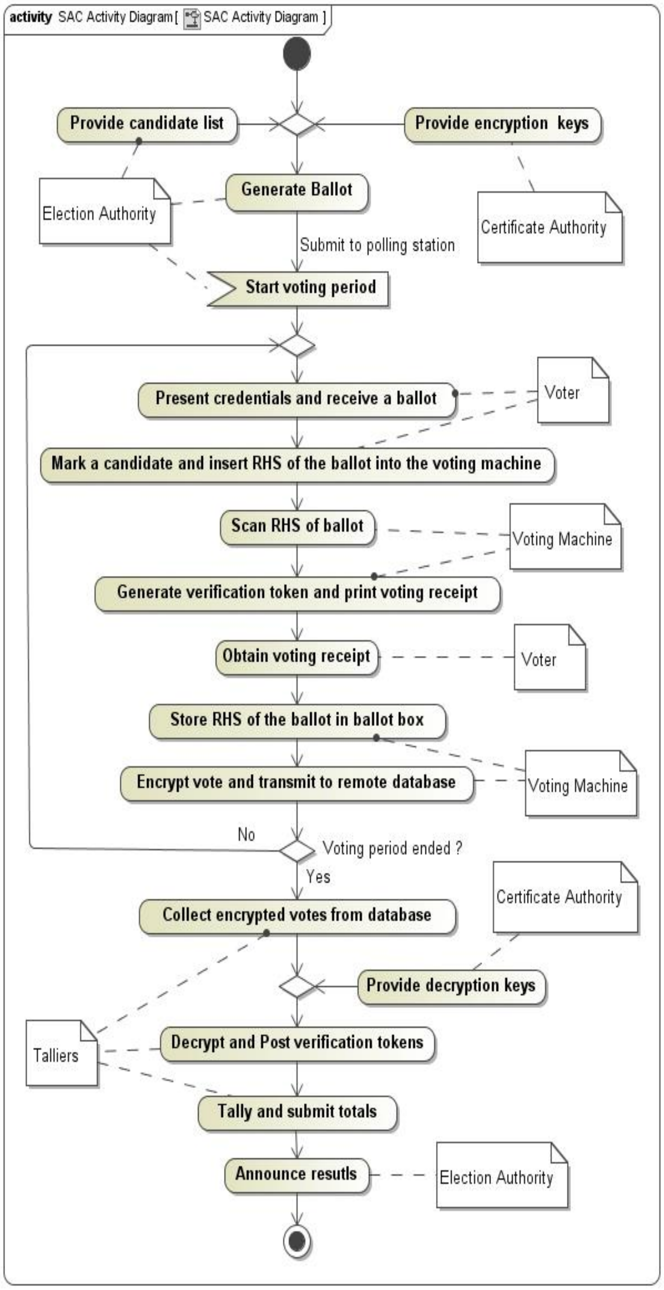

activity SAC Activity Diagram[SAC Activity Diagram]
Provide candidate list
Provide encryption keys
Election Authority
Generate Ballot
Certificate Authority
Submit to polling station
Start voting period
Present credentials and receive a ballot
Voter
Mark a candidate and insert RHS of the ballot into the voting machine
Scan RHS of ballot
Voting Machine
Generate verification token and print voting receipt
Obtain voting receipt
Voter
Store RHS of the ballot in ballot box
Encrypt vote and transmit to remote database
Voting Machine
No
Voting period ended ?
Yes
Collect encrypted votes from database
Certificate Authority
Provide decryption keys
Talliers
Decrypt and Post verification tokens
Tally and submit totals
Announce results
Election Authority

Fig.3 SAC e-voting scheme procedures

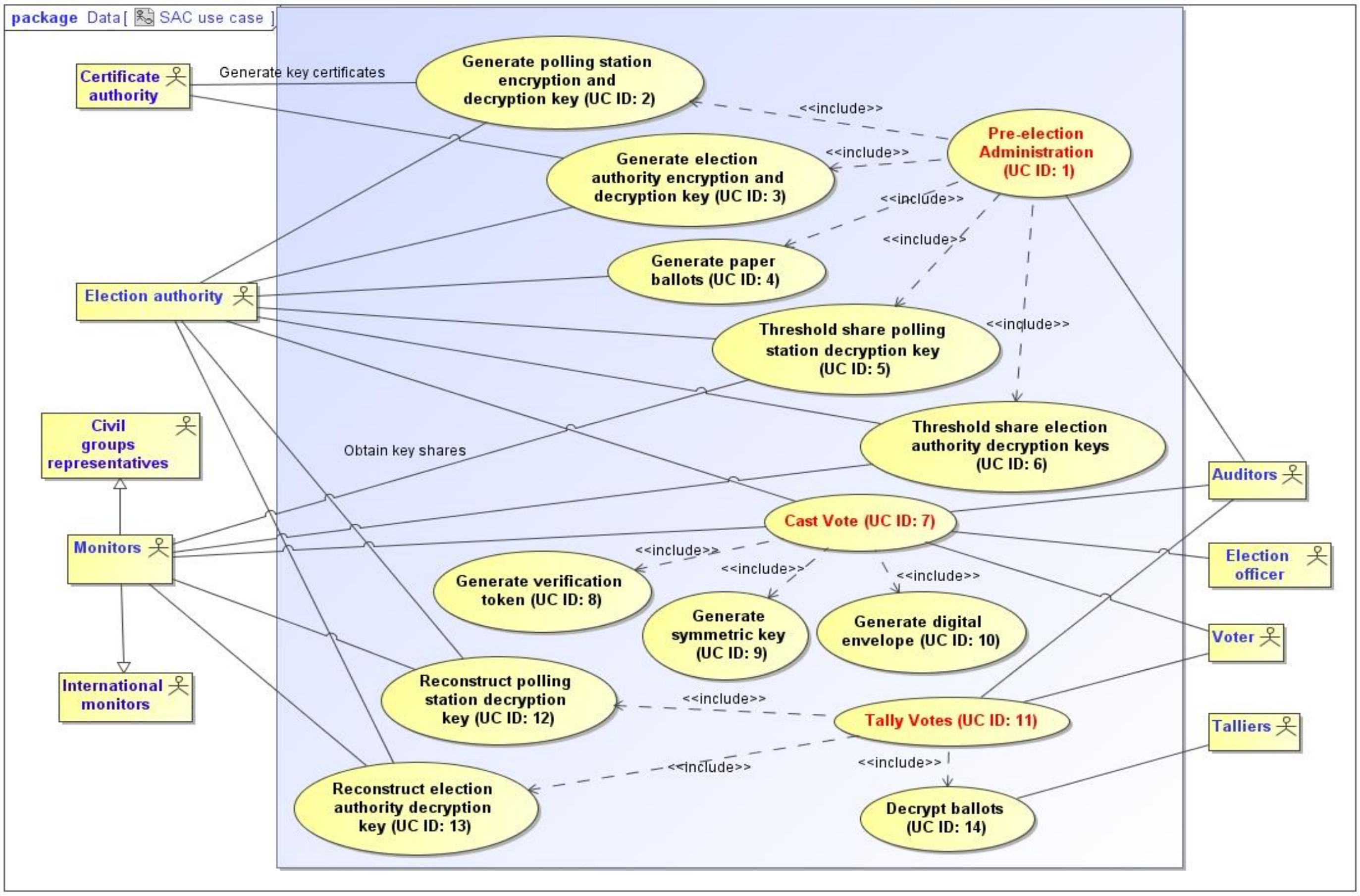

package Data [SAC use case]
Certificate authority
Generate key certificates
Generate polling station encryption and decryption key (UC ID: 2)
<<include>>
Pre-election Administration (UC ID: 1)
Generate election authority encryption and decryption key (UC ID: 3)
<<include>>
<<include>>
<<include>>
Generate paper ballots (UC ID: 4)
Election authority
Threshold share polling station decryption key (UC ID: 5)
<<include>>
Threshold share election authority decryption keys (UC ID: 6)
Civil groups representatives
Obtain key shares
Auditors
Monitors
Cast Vote (UC ID: 7)
<<include>>
<<include>>
<<include>>
Election officer
Generate verification token (UC ID: 8)
Generate symmetric key (UC ID: 9)
Generate digital envelope (UC ID: 10)
Voter
International monitors
Reconstruct polling station decryption key (UC ID: 12)
<<include>>
Tally Votes (UC ID: 11)
Talliers
<<include>>
<<include>>
Reconstruct election authority decryption key (UC ID: 13)
Decrypt ballots (UC ID: 14)

Fig. 4 The SAC use case diagram

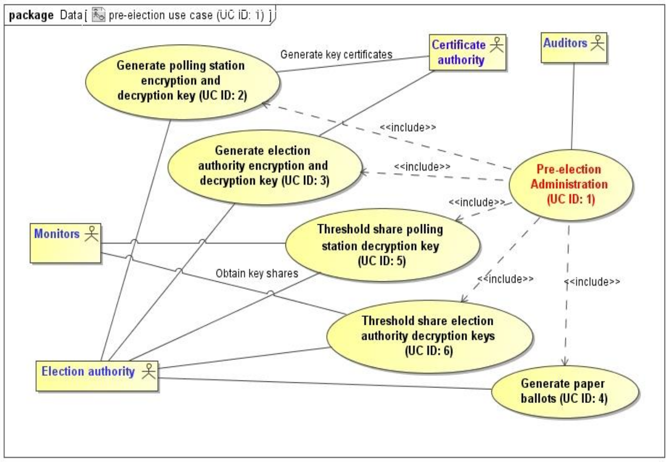

package Data [pre-election use case (UC ID: 1)]
Generate key certificates
Certificate authority
Auditors
Generate polling station encryption and decryption key (UC ID: 2)
<<include>>
Generate election authority encryption and decryption key (UC ID: 3)
<<include>>
Pre-election Administration (UC ID: 1)
<<include>>
Monitors
Threshold share polling station decryption key (UC ID: 5)
Obtain key shares
<<include>>
<<include>>
Threshold share election authority decryption keys (UC ID: 6)
Election authority
Generate paper ballots (UC ID: 4)

Fig. 5 Pre-Election Administration use case

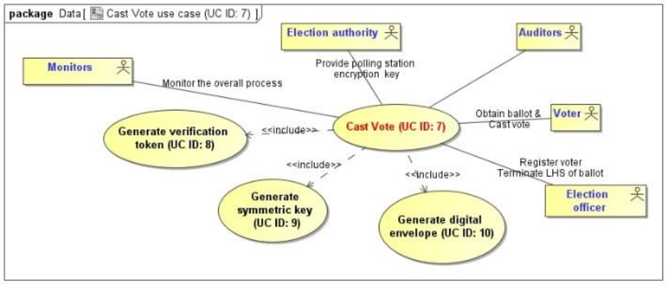

package Data [Cast Vote use case (UC ID: 7)]
Election authority
Auditors
Monitors
Provide polling station encryption key
Monitor the overall process
Obtain ballot & Cast vote
Voter
Generate verification token (UC ID: 8)
<<include>>
Cast Vote (UC ID: 7)
<<include>>
<<include>>
Register voter
Terminate LHS of ballot
Generate symmetric key (UC ID: 9)
Generate digital envelope (UC ID: 10)
Election officer

Fig. 6 Cast Vote use case

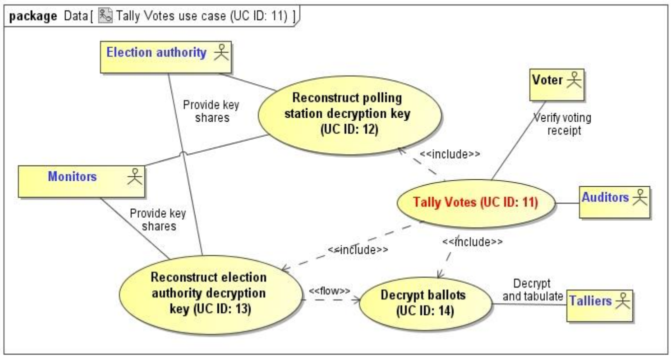


Fig. 7 Tally Votes use case

The use case diagram that incorporates all the use cases of the SAC e-voting scheme is shown in Fig. 4. The three main use cases are Pre-election Administration (UC ID: 1) shown in Fig. 5, Cast Vote (UC ID: 7) shown in Fig. 6 and Tally Votes (UC ID: 11) shown in Fig. 7. The diagram shown in Fig. 4 provides an overview of the whole SAC e-voting scheme and exhibits its full functionality and the tasks that each actor is involved in. Samples of the developed activity diagrams are shown in Fig. 8 & Fig. 9. The full activity diagrams, use cases and procedures, which describe in details the different scenarios of the SAC e-voting scheme are presented in [42].

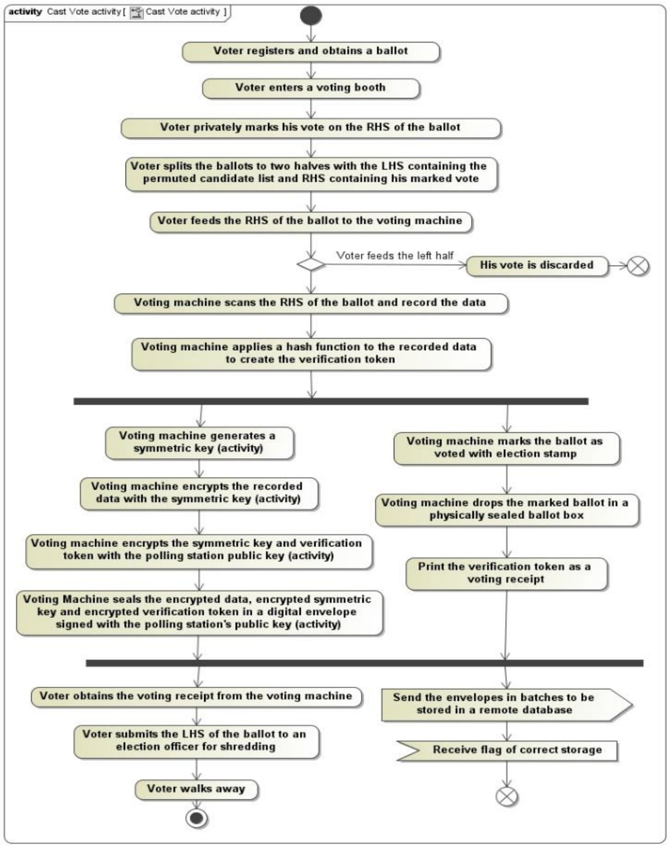

activity Cast Vote activity [Cast Vote activity]
Voter registers and obtains a ballot
Voter enters a voting booth
Voter privately marks his vote on the RHS of the ballot
Voter splits the ballots to two halves with the LHS containing the permuted candidate list and RHS containing his marked vote
Voter feeds the RHS of the ballot to the voting machine
Voter feeds the left half
His vote is discarded
Voting machine scans the RHS of the ballot and record the data
Voting machine applies a hash function to the recorded data to create the verification token
Voting machine generates a symmetric key (activity)
Voting machine encrypts the recorded data with the symmetric key (activity)
Voting machine encrypts the symmetric key and verification token with the polling station public key (activity)
Voting Machine seals the encrypted data, encrypted symmetric key and encrypted verification token in a digital envelope signed with the polling station's public key (activity)
Voting machine marks the ballot as voted with election stamp
Voting machine drops the marked ballot in a physically sealed ballot box
Print the verification token as a voting receipt
Voter obtains the voting receipt from the voting machine
Voter submits the LHS of the ballot to an election officer for shredding
Voter walks away
Send the envelopes in batches to be stored in a remote database
Receive flag of correct storage

Fig. 8 Cast Vote activity diagram

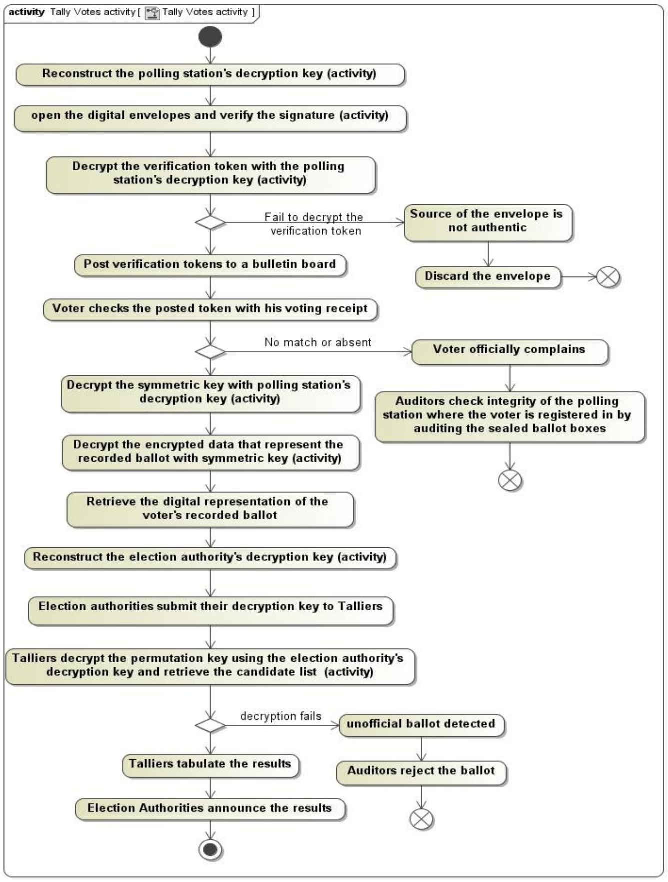

activity Tally Votes activity [Tally Votes activity]
Reconstruct the polling station's decryption key (activity)
open the digital envelopes and verify the signature (activity)
Decrypt the verification token with the polling station's decryption key (activity)
Fail to decrypt the verification token
Source of the envelope is not authentic
Discard the envelope
Post verification tokens to a bulletin board
Voter checks the posted token with his voting receipt
No match or absent
Voter officially complains
Auditors check integrity of the polling station where the voter is registered in by auditing the sealed ballot boxes
Decrypt the symmetric key with polling station's decryption key (activity)
Decrypt the encrypted data that represent the recorded ballot with symmetric key (activity)
Retrieve the digital representation of the voter's recorded ballot
Reconstruct the election authority's decryption key (activity)
Election authorities submit their decryption key to Talliers
Talliers decrypt the permutation key using the election authority's decryption key and retrieve the candidate list (activity)
decryption fails
unofficial ballot detected
Auditors reject the ballot
Talliers tabulate the results
Election Authorities announce the results

Fig. 9 Tally Votes activity diagram

VI. SAC E-Voting Scheme Details and Security Properties

Table V presents the notations used to describe the SAC e-voting scheme. The scheme will be described in details through its three main stages hereafter.

Table V
Scheme notations

Notation	Description
PK	Public key of an entity
SK	Private key of an entity
L	List of candidates
N	Permutation key
V	Vote cast by a voter
EPK	Encryption of a message with a public key
DSK	Decryption of a cipher text with a secret key. Note that DSK (EPK(V)) = V
TH	Threshold secret sharing scheme
TH^{-1}	Reconstruction of a secret from its shares
SHA-1	One way hash function
S	Symmetric key
ES	Encryption with symmetric key
DS	Decryption with symmetric key
Sign	Digital signature of a digital envelope

***Stage 1*:**

a. Generate polling station's key set { $PK_{P,}$ SK_P} where PK_P is the encryption key (public key) and SK_P is the decryption key (private key)

b. Generate election authority's key set { $PK_{A,}$ SK_A} where PK_A is the encryption key (public key) and SK_A is the decryption key (private key)

c. Generate election authority's digital signature key set {PK_S, SK_S} where SK_S is the signing key and PK_S is the signature verifying key.

d. Prepare an unordered list of candidates "L" where L >= 2

e. Perform "N" permutations on the candidate list => N (L)

f. Encrypt the permutation key "N" with the election authority's encryption key => EPK_A(N)

g. Represent the cipher EPK_A(N) as a two dimensional bar code

h. Print the permuted list of candidates N (L) on the LHS of the ballot and the two dimensional bar code representation of EPK_A(N) on the RHS of the ballot

i. Threshold share polling station's decryption key and election authority decryption key => TH (SK_P), TH (SK_A)

j. Destroy the election authority's encryption key PK_A

***Stage 2*:**

a. Record the data "V" that represent the vote and barcode

b. Apply SHA-1 hash function to the recorded data "V" and present the digest as the verification token =>SHA-1 (V)

c. Generate a symmetric key "S"

d. Encrypt the data "V" with the symmetric key => ES (V)

e. Encrypt the symmetric key and verification token with the polling station's encryption key
   => EPK_P (S), EPK_P (SHA-1(V))
f. Sign the encrypted data, encrypted symmetric key and encrypted verification token signed with the election authority signing key
   => Sign (SK_S) [{ES(V), EPK_P (S), EPK_P (SHA-1 (V))}]

***Stage 3*:**

a. Verify the digital signature of the digital envelopes to validate its source and its authenticity using the election authority signature verification key
   => Sign (PK_S) [{ES(V), EPK_P (S), EPK_P (SHA-1 (V))}]
b. Reconstruct the polling station's decryption key from its shares => TH^{-1} (SK_P)
c. Retrieve the digital envelopes after signature verification
   => {ES(V), EPK_P (S), EPK_P (SHA-1 (V))}
d. Decrypt the verification token with the polling station decryption key => DSK_P [EPK_P [SHA-1 (V)]
e. Post the verification token SHA-1(V) to a bulletin board
f. Decrypt the symmetric key with polling station's decryption key => DSK_P (EPK_P (S))
g. Decrypt the data "V" that represent the vote with the symmetric key => DS (ES(V))
h. Retrieve the digital representation of the vote "V" and the encrypted permutation key EPK_A(N)
i. Reconstruct the election authority's decryption key from its shares => TH^{-1} (SK_A)
j. Decrypt the permutation key "N" with the election authority's decryption key => DSK_A (EPK_A(N))
k. Retrieve the candidate list "L" by rearranging the permuted candidate list N(L) using the retrieved permutation key "N"
l. Record the voter's choice "V" that correspond to his chosen candidate in the candidate list "L"

The SAC e-voting scheme follows the functional requirements of electronic voting as well as the constitutional requirements of the chosen case study Egypt. The Scheme is compared by the definition of its security properties against the desirable properties of e-voting. This comparison of security properties will clarify the potentials of the SAC e-voting scheme and further support the claimed aspects of the scheme. An exhaustive set of definitions of the security properties for electronic voting schemes and systems can be found [1].

The new scheme satisfies eligibility and authentication properties; only voters satisfying the voter's requirements are listed in the registration databases, and only voters listed in the registration databases are allowed to vote.

The scheme satisfies uniqueness and non-reusability; no voter should be able to vote more than once, therefore a voter gets only one ballot at the polling station provided that the electoral roll is in effect.

The SAC e-voting scheme is secure in terms of providing tamper resistance of votes and in turn satisfying confidentiality and integrity of votes as well as non-repudiation of their origin. This is done by sealing the encrypted digital representation of the vote in a digital envelope signed with the election authority private signing key. Thus no one can change or duplicate someone else's vote. Secure audit logs should be achieved on the remote database server where the digitally signed envelopes are stored, to prevent undetected tampering or deletion, which will in turn further satisfy the integrity property.

The privacy property is satisfied through the use of voting booths present at the polling stations. Convenience property is satisfied since the voter would cast the vote with minimal equipment and skills without having to learn too complex techniques in order to vote.

Voters should be able to possess a general understanding of the whole process thus satisfying transparency property. After voting, the voter is not involved in any other post vote process satisfying walk away property. The simple ballot structure and the ballot submission and verification mechanisms should not raise any disputes.

The SAC e-voting scheme does not have assumptions and requirements such as anonymous channels (Mixnets), that may be difficult to implement on a large scale, which makes it an attractive candidate for pilot implementation. Practicality will be further satisfied when the scheme is implemented to verify its scalability and efficiency. The fairness property is satisfied through the threshold secret sharing scheme, which ensures that no one can learn the outcome of the elections before the announcement of the tally.

The incoercibility property is conditionally satisfied since the voting receipt prevents the voter from proving to others how he voted. Therefore a coercer will not force the voter to vote in a certain way nor will the coercer bribe a voter since there is no means to prove that the voter followed the coercer's intentions. Fully satisfying incoercibility property is hard since a coercer may force a voter to abstain from voting but a coercer cannot impersonate a voter due to the various pre-voting checks at the polling station.

Receipt freeness property is not satisfied as it is traded off with verifiability in which the voting receipts play an important role in both individual and universal verifiability. Voters shall be able to verify that their votes are correctly included in the final tally, as well as auditors can further verify that the whole voting and tallying processes are correct.

The scheme satisfies the flexibility property and can be used for several types of elections such as approval voting, Borda count voting, STV, and Condorcet voting. The scheme has no special requirements that limit its implementation and use, therefore it should be affordable in terms of hardware and maintenance. This intern satisfies cost effectiveness and feasibility properties.

Verifiable Participation property ensures that it is possible to find out whether a particular voter has participated in the election by casting a ballot or not. This can be easily verified by checking the polling station registry book that voters sign in order to obtain a ballot.

Efficiency property focuses on avoiding too many steps to reach the goal of the voting process. The design of the e-voting scheme avoided the use of too complex techniques such as anonymous channels to provide scheme simplicity through the use of public key cryptography, which in turn increases its efficiency relative to other schemes that use anonymous channels. Scheme implementation will further verify the efficiency property.

The use of the symmetric key encryption adds two desirable properties; more cryptographic security to the electronic records to prevent tampering and provides light weight cryptography, rather than using the polling station's public key to encrypt the electronic record of the voter's choice. A voting scheme has to be scalable with respect to storage, computation, and communication needs to accommodate larger number of voters. Again the scheme implementation will further verify the *scalability* property.

Verifiability property is satisfied in a variety of ways; a voter can use his voting receipt containing his ballot's verification token to verify that his vote was included in the final tally. Verification tokens can also be used by election monitors and auditors during the opening phase of the signed digital envelopes to verify the correct decryption of ballot's electronic records. This can be done by re-applying the hash function to the decrypted electronic record of the ballot. After decrypting the verification token and the electronic record of the ballot, the auditors then match the verification token obtained from the envelope with the newly generated hash digest of electronic record to verify its integrity. Digital signature of digital envelopes containing valid votes also serve to satisfy verifiability property.

Verification checks of the whole voting process can be done using a set of marked ballots than are introduced to the voting machines at any time during the voting period. This process is performed with a complete voting experience and can be conducted by international monitors, auditors, civil group representatives and candidates' representatives to verify the whole voting and tallying processes. The ballots may be marked with a set of election authorities' digital certificates with key pairs that could be excluded from final tally yet provide strong proof of correct operation before and during the election period.

A brief overview of auditing is discussed hereafter. The procedure of information security audit consists of 4 phases: the planning phase, the implementation phase, the reporting phase, and the improvement phase [43, 44]. In the planning phase, the plan for document audit and on-the-spot audit is made by extracting necessary audit items according to the purpose of each audit. The specific work of this phase includes identifying where the necessary data exist, determining the range of audit, and so on. Thus, the amount of audit work greatly varies according to the size of the audit target. In the implementation phase, each item is audited under the audit plan. In the reporting phase, the results of the audit in the implementation phase are documented and reported to the election authorities to take proper actions where necessary. In the improvement phase, a plan is made in order to improve the audit items that have been judged as incompatible to the audit criteria.

Audit-ability property is also satisfied in a variety of ways. Quick and verifiable tallies have to be generated when the election period is over, while incorporating a variety of audit mechanisms facilitated to auditors with additional privileges. From an auditability standpoint, the presence of the paper ballots preserve a separate physical copy apart from the electronic digital records of the ballots, which allows matching of those paper ballots and the electronic records with the help of the election stamps. In addition, the paper records remain available in case of systemic failure of the electronic records or if a manual count is ever needed. Also, Part of a system audit of a polling stations consists of log files residing in voting machines, in which events that occur in each component are recorded. These log files contain data regarding the opening and closing times of the voting machine, the numbers of votes cast, the number of receipts generated etc. These log files are subject to inspection if auditing of a specific voting machine or polling station is ever required.

On the other hand, the paper audit trail (ballots present in the physically sealed ballot boxes) enables an entirely independent check. It can verify that the votes were included and tabulated accurately, that the visible trace of voter's intent as reflected in the ballot agrees with the digital records, and more important, that the winners reported by the voting system are the winners that a full hand count of the audit trail would reveal. A risk-limiting audit process serves to verify the correspondence between the paper records (RHS of the ballots) and the electronic records (digital representation of the vote data). Risk-limiting audits are widely considered as best practice for election audits.

The SAC e-voting scheme may be expanded to accommodate voting for eligible people living abroad and expats. A voting day will differ from one country to another due to the different time zones that those voters are experiencing. So there has to be a way for those voters to vote during the voting period of the chosen case study Egypt. Voting from polling stations residing inside embassies and consulates would be justified in critical political elections, due to concerns related to security and privacy and also to combat bribe and coercion. The voting booths may connect to a database server that will collect the digital envelopes after each vote cast through a secure connection via a tunneling technique. Another way for vote collection is that the digital envelopes may be sent after each country's voting period for storage on a secured remote database server. Those digital envelopes may then be processed after the voting period is over. Paper ballots will be stored in embassies for manual recount if needed.

To justify voting in polling stations for the SAC e-voting scheme whether voters reside in Egypt or live abroad, security and privacy need to be maintained. Remote internet e-voting from home or at work is inherently coercible as there is no guaranteed privacy during ballot casting. Remote Internet voting systems suffer from many security problems which rely on the clients, the servers, and the network connections. Distributed denial of service (DDoS) attacks and malware infections still belong to the most challenging security issues.

Penetration attacks target the client or server directly whereas denial of service attacks target and interrupt the communications link between the two. Penetration attacks involve the use of a delivery mechanism to transport a malicious payload to the target host in the form of a Trojan horse or remote control program. Once executed, it can spy on ballots, prevent voters from casting ballots, or, even worse, modify the ballot according to its instructions. Remote control software may compromise the secrecy and integrity of the ballots by those monitoring the host's activity.

Remote internet voting will also have to contend with an attack known as spoofing; luring unwitting voters to connect to an imposter web site instead of the actual election servers. While technologies such as secure socket layer (SSL) and digital certificates are capable of distinguishing legitimate servers from malicious ones, it is infeasible to assume that all voters will have these protections functioning properly on their home or work computers, and, in any event, they cannot fully defend against all such attacks. Successful spoofing can result in the undetected loss of a vote should the user send his ballot to a fake voting site. Even worse, the imposter site can act as a phishing site between a voter and the real site, and may steal the voter's credentials and vote on his behalf, or change the vote itself then send it to the real site.

In principle, poll site voting is much less susceptible than remote internet voting to the previously mentioned attacks. The software and hardware on the voting machines would be controlled and supervised by elections officials, and would be configured so as to prevent communication with any Internet hosts except the proper election servers through secure communication channels.

VII. Conclusion

The work presented here is concluded by briefly stating the main features of the *SAC* e-voting scheme which makes it an excellent candidate for direct implementation and deployment in real world. The proposed scheme is Secure due to the use of several cryptographic encryption algorithms which provides enhanced level of security as well as privacy and vote anonymity. The proposed Scheme is Audit-able with the use of risk limiting audits and parallel test of voting machines. The proposed scheme has no special requirements relative to the voter which would allow the majority of voters in developing countries to accept and participate in elections thus increasing the voters' turnout. The use of light weight yet effective cryptography would facilitate the replacement of conventional voting methods with electronic voting by using low cost computer hardware.

VIII. Future Work

The *SAC* e-voting scheme should be expanded to accommodate all aspects of a full e-voting experience to form a fully functional electronic voting system. The *SAC* e-voting scheme should undergo pilot implementation and functional tests should be carried out to verify the efficiency, scalability and practicality of the scheme. A full mathematical model of the key security properties of the scheme and its resistance to well-known security attacks is the subject of an ongoing research.

References

[1] Mursi, M. F., Assassa, G. M., Abdelhafez, A., & AboSamra, K. M. On the Development of Electronic Voting: A Survey. In International Journal of Computer Applications, 61(16), 2013, pp. 1-11.

[2] Chaum D. Blind signature system, Advances in cryptology –CRYPTO '83. Plenum Press; 1984. pp. 153.

[3] Fujioka, A., Okamoto, T., & Ohta, K. A practical secret voting scheme for large scale elections. In Advances in Cryptology—AUSCRYPT'92, Springer Berlin Heidelberg, 1993, pp. 244-251.

[4] Okamoto, T. An electronic voting scheme. In Advanced IT Tools, Springer US, 1996, pp. 21-30.

[5] Cohen, J. D., & Fischer, M. J. A robust and verifiable cryptographically secure election scheme. In Foundations of Computer Science, 26th Annual Symposium on, IEEE, 1985, pp. 372-382.

[6] Benaloh, J. C., & Yung, M. Distributing the power of a government to enhance the privacy of voters. In Proceedings of the fifth annual ACM symposium on Principles of distributed computing, ACM, 1986, pp. 52-62.

[7] Benaloh, J., & Tuinstra, D. Receipt-free secret-ballot elections. In Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, ACM, 1994, pp. 544-553.

[8] Cramer, R., Franklin, M., Schoenmakers, B., & Yung, M. Multi-authority secret-ballot elections with linear work. In Advances in Cryptology—EUROCRYPT'96, Springer Berlin Heidelberg, 1996, pp. 72-83.

[9] Cramer, R., Gennaro, R., & Schoenmakers, B. A Secure and Optimally Efficient Multi-Authority Election Scheme. In Advances in Cryptology—EUROCRYPT'97, Springer Berlin Heidelberg, 1997, pp. 103-118.

[10] Baudron, O., Fouque, P. A., Pointcheval, D., Stern, J., & Poupard, G. Practical multi-candidate election system. In Proceedings of the twentieth annual ACM symposium on Principles of distributed computing, ACM, 2001, pp. 274-283.

[11] Feng, C., Xin, Y., Yang, Y., & Zhu, H. Multi-integer Somewhat Homomorphic Encryption Scheme with China Remainder Theorem. WSEAS transactions on computers, 14(1), 2015, pp. 1-13.

[12] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2), 1981, pp. 84-90.

[13] Sako, K., & Kilian, J. Receipt-free mix-type voting scheme. In Advances in Cryptology—EUROCRYPT'95. Springer Berlin Heidelberg, 1995, pp. 393-403.

[14] Hirt, M., & Sako, K. Efficient receipt-free voting based on homomorphic encryption. In Advances in Cryptology—EUROCRYPT 2000, Springer Berlin Heidelberg, 2000, pp. 539-556.

[15] Neff, C. A. A verifiable secret shuffle and its application to e-voting. In Proceedings of the 8th ACM conference on Computer and Communications Security, ACM, 2001, pp. 116-125.

[16] Jakobsson, M., Juels, A., & Rivest, R. L. Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking. In USENIX security symposium, 2002, pp. 339-353.

[17] Kusters, R., Truderung, T., & Vogt, A. Formal analysis of chaumian mix nets with randomized partial checking. In Security and Privacy (SP), IEEE Symposium on, IEEE, 2014, pp. 343-358.

[18] Bernhard, D., Neumann, S., & Volkamer, M. Towards a practical cryptographic voting scheme based on malleable proofs. In E-Voting and Identify, Springer Berlin Heidelberg, 2013, pp. 176-192.

[19] Chase, M., Kohlweiss, M., Lysyanskaya, A., & Meiklejohn, S. Verifiable elections that scale for free. In Public-Key Cryptography–PKC 2013, Springer Berlin Heidelberg, 2013, pp. 479-496.

[20] Chaum, D., Ryan, P. Y., & Schneider, S. A practical voter-verifiable election scheme. In Proceedings of the 10th European conference on Research in Computer Security, Springer-Verlag, 2005, pp. 118-139.

[21] Ryan, P. Y. A variant of the Chaum voter-verifiable scheme. In Proceedings of the 2005 Workshop on Issues in the Theory of Security, ACM, 2005, pp. 81-88.

[22] Ryan, P. Y., & Schneider, S. A. Prêt à voter with re-encryption mixes. In Proceedings of the 11th European conference on Research in Computer Security, Springer-Verlag, 2006, pp. 313-326.

[23] Ryan, P. Y. Prêt à Voter with Paillier encryption. Mathematical and Computer Modelling, 48(9), 2008, pp. 1646–1662.

[24] Ryan, P. Y., Bismark, D., Heather, J., Schneider, S., & Xia, Z. Prêt à voter: a voter-verifiable voting system. Information Forensics and Security, IEEE Transactions on, 4(4), 2009, pp. 662-673.

[25] Demirel, D., Henning, M., van de Graaf, J., Ryan, P. Y., & Buchmann, J. Prêt à voter providing everlasting privacy. In E-Voting and Identify, Springer Berlin Heidelberg, 2013, pp. 156-175.

[26] Khader, D., Ryan, P., & Tang, Q. Proving Prêt à Voter Receipt Free Using Computational Security Models. USENIX Journal of Election Technology and Systems (JETS), 1(1), 2013, pp. 62-81.

[27] Carback, R. T., Popoveniuc, S., Sherman, A. T., & Chaum, D. Punchscan with independent ballot sheets: Simplifying ballot printing and distribution with independently selected ballot halves. In Proceedings of the 2007 International Association for Voting Systems Sciences IAVoSS, workshop on trustworthy elections WOTE, 2007.

[28] Essex, A., Clark, J., Carback, R., & Popoveniuc, S. Punchscan in practice: an E2E election case study. In Proceedings of the 2007 International Association for Voting Systems Sciences IAVoSS, workshop on trustworthy elections WOTE, 2007.

[29] Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., & Vora, P. Scantegrity: End-to-end voter-verifiable optical-scan voting. Security & Privacy, IEEE, 6(3), 2008, pp. 40-46.

[30] Chaum, D., Carback, R. T., Clark, J., Essex, A., Popoveniuc, S., Rivest, R. L. & Vora, P. Scantegrity II: End-to-end verifiability by voters of optical scan elections through confirmation codes. Special Issue on Electronic Voting, Information Forensics and Security, IEEE Transactions on, 4(4), 2009, pp. 611-627.

[31] Sherman, A. T., Fink, R. A., Carback, R., & Chaum, D. Scantegrity III: Automatic Trustworthy Receipts, Highlighting Over/Under Votes, and Full Voter Verifiability. In In Proceedings of the 2011 conference on Electronic voting technology, workshop on trustworthy elections EVT/WOTE'11, USENIX Association, 2011, pp. 7-7.

[32] Adida, B., & Rivest, R. L. Scratch & vote: self-contained paper-based cryptographic voting. In Proceedings of the 5th ACM workshop on Privacy in electronic society, 2006, pp. 29-40.

[33] Randell, B., & Ryan, P. Y. Voting Technologies and Trust. Security & Privacy, IEEE, 4(5), 2006, pp. 50-56.

[34] Rivest, R. L., & Smith, W. D. Three voting protocols: ThreeBallot, VAV, and twin. In Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology EVT'07, USENIX Association, 2007, pp. 16-16.

[35] Araújo, R., Custódio, R., Wiesmaier, A., & Takagi, T. (2006). An electronic scheme for the Farnel paper-based voting protocol. ACNS'06.

[36] Araújo, R. Improving the Farnel, Threeballot, and Randell-Ryan Voting Schemes. IACR Cryptology ePrint Archive (82), 2008.

[37] Araújo, R., Custódio, R. F., & Van De Graaf, J. A verifiable voting protocol based on Farnel. In Towards Trustworthy Elections, Springer Berlin Heidelberg, 2010, pp. 274-288.

[38] Chor B, Goldwasser S, Micali S, Awerbuch B. Verifiable secret sharing and achieving simultaneous broadcast. In: Proceedings of IEEE foundation of computer science 1985. p. 335–44.

[39] Schoenmakers B. A simple publicly verifiable secret sharing scheme and its applications to electronic voting. In: Advances in cryptology – CRYPTO '99. LNCS, vol. 1666. Springer-Verlag, 1999. p. 148–64.

[40] Heather, J., & Lundin, D. The Append-Only Web Bulletin Board. In Formal Aspects in Security and Trust (FAST'08), 5th International Workshop, Springer Berlin Heidelberg, 2009, pp. 242-256.

[41] Culnane, C., & Schneider, S. A peered bulletin board for robust use in verifiable voting systems. In Computer Security Foundations Symposium (CSF), IEEE 27th, IEEE, 2014, pp. 169-183.

[42] AboSamra, K. M. A practical, secure and audit-able e-voting system. Ph.D. thesis in preparation, to be submitted to the faculty of engineering, Benha University, Egypt, in 2016.

[43] Satoh, N., & Kumamoto, H.h Estimation Model of Labor Time at the Information Security Audit and Standardization of Audit Work by Probabilistic Risk Assessment, International journal of computers, North Atlantic University Union (NAUN), 2009.

[44] Satoh, N., & Kumamoto, H. Viewpoint of Probabilistic Risk Assessment in Information Security Audit, International journal of computers, North Atlantic University Union (NAUN), 2009.

Mona F.M. Mursi is currently a professor in the Computer Section of the EE department, Faculty of Engineering, Benha University, Egypt, since 1980. She received the B.Sc. degree from the College of Engineering, Cairo University in 1965 and was assistant professor at the EE Department, Cairo University. She earned a Masters degree in 1967 from University of Manchester, UK, as well as a second Masters degree from the University of Missouri at Columbia, MO, USA in 1969, and earned a Ph.D. from the University of Missouri in 1972. Prof. Mursi also headed the IT department of King Saud University, S.A. during the period 1988-1994 and again during the period 2001-2009. Prof. Mursi also worked in industry in the US (1972-1979): at Tektronix Inc., USA and Calma Inc. Prof. Mursi worked as a consultant in the computer department. Areas of research interest are computer and network security, and bioinformatics.

Ghazy Assassa works at the Faculty of Engineering, Benha University, Egypt. Since 2009, he is advisor to university president for IT and the university Chief Information Officer. He received his B.Sc. degree from the Faculty of Engineering, Cairo University in 1967 and earned his Masters and Ph.D. degrees from Lyon University, France in 1972 and 1976, respectively. He joined CNRS in France in 1976 and worked in the department of Computer Science at King Saud University (2002-2009). Professor Ghazy Assassa has more than 60 research papers in the areas of CFD, modeling and simulation, steganography, software engineering, renewable energy. He is Editor-in-Chief of the International Journal of Service Science, Management, Engineering, and Technology.

Ahmed A. AbdelHafez received the B.Sc. and M.Sc. in Electrical Engineering from Military Technical College (MTC) in 1990, 1997 respectively, and his Ph.D. from School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, Canada in 2003. Dr. Abdel-Hafez is the head of the Cryptography Research Center (CRC), Egypt where he is leading many applied researches in communication security field. He is a visiting lecturer in Communication Dept. MTC, and other universities in Egypt. Dr. Abdel-hafez published more than 40 papers in specialized conferences and periodicals. His research interests include wireless networks and data security, mathematical cryptography and provable security.

Kareem M. AboSamra received his Masters degree in 2010 from the Arab Academy for Science & Technology, Cairo, Egypt. He joined the doctoral program in 2012 at the faculty of computer engineering, Benha University, Giza, Egypt. Eng. AboSamra is an Assistant Lecturer at the faculty of electrical and computer engineering, the Higher Technological Institute, 6th of October, Egypt. Eng. AboSamra has presented his research at the International Journal of Computer Applications, and the Ain-Shams journal of electric engineering, Cairo, Egypt. His research interests include information security, cryptography and network security.