# Eligibility Verifiability in Untrustworthy Voting Environments

Submitted by

Voke Augoye

for the degree of Doctor of Philosophy

of the

Royal Holloway, University of London

2021

## Declaration

I, Voke Augoye, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed ..... (Voke Augoye)

Date:

*In loving memory of*

*Mr Charles Augoye 1943-2010*

*Mrs Catherine Enughe Augoye 1950-2018*

# Abstract

For citizens of a nation to have a faith in its democracy, they must be assured of the integrity of the ballots, secrecy of the ballots and reliability of the outcome of the election after ballots have been counted.

In most electronic voting schemes (EVS) both in the literature and practical schemes deployed, trust is placed on various entities in the voting process to act honestly, if this trust assumptions hold, then the security requirements of an e-voting scheme is achieved. However, these trust assumptions may not always hold in practical deployment due to threats that exist in the real world that may not have been considered as at time of design.

Eligibility verifiability is an important security requirement of a Voting Scheme, if this requirement is not satisfied then ballot stuffing, ineligible voting and impersonation of legitimate voter cannot be prevented. These forms of electoral fraud have been widely reported in many elections around the world.

End-to-End Verifiability in EVS, gives voters assurance their votes have been cast-as-intended, recorded-as-cast and counted-as-recorded. However, in many End-to-End EVS, voter authentication is done externally to the voting scheme. An example is the Prêt-à-Voter schemes [62] that relies on poll-workers to authenticate voters using non technical mean. If these trust assumptions are broken, then the security requirements are not met. This breach in trust gives rise to the notion of an untrustworthy voting environment.

In this thesis we define two untrustworthy environments and the threats in these environments, making them unsuitable for elections. We propose Electronic Voting Schemes that addresses the threats that exists in these Untrustworthy Environments.

Finally, we incorporate our generic schemes into existing Electronic Voting Schemes, do a security analysis and show that our schemes achieves eligibility verifiability as well as other security requirements of an EVS.

# Acknowledgement

First of all I want to acknowledge God almighty for seeing me through this process. I want to thank my Supervisor Dr. Allan Tomlinson for his constant support, encouragement and guidance through out my time at Royal Holloway,University of London right from my Msc thesis and through this PhD process. I also want to thank Professor Keith Martins who stepped in at the last minute to supervise me.

I want to specially thank my amazing wife for being a pillar of support and source of constant motivation these past years. I particularly want to thank her for continuously pulling me trough the most difficult parts of this research programme.

To my amazing siblings, Rume, Onome and Otome, I cannot thank you guys enough for all the support, both financially and spiritually through this research process. I am truly grateful to my cousin Kobi and his family for their constant support and care, I couldn't have done it with you guys. Thanks must also go to my uncles, aunties, friends and external family for their constant prayers and love shown to me all these years.

To my late parents, I don't know where to begin, but I want to say a very big thank you for all your love, care and financial support over the years up till today because without the finances I wouldn't have been able to survive this PhD programme. The thought of making you proud has motivated and pushed me through the most trying periods of this research programme.

Last but not the least, I want to give a very special thanks to the Department of Mathematics & Information Security Group for the College Oversea Scholarship granted to me, without that scholarship I wouldn't have been able to attend this PhD programme.

# Contents

# List of Figures

# List of Tables

# List of Notation

| | |
|---|---|
| $ID_X$ | Entity X's identifier. |
| $X_{PV}$ | Entity X's private key. |
| $X_{PB}$ | Entity X's public key. |
| $Cert_X$ | Entity X's public key certificate. |
| $K_{X,Y}$ | Secret key shared between X and Y. |
| $SK$ | Session Key |
| $(M)_K$ | Encryption of message M with key K. |
| $MAC_K(M)$ | Message Authentication Code on M with key K. |
| $S_X(M)$ | Sign message M with X's signing key. |
| $X\|\|Y$ | Concatenation of X and Y. |

# List of Abbreviations

| | |
|---|---|
| API | Application program Interface |
| CA | Client Application |
| EA | Election Authority |
| EVS | Electronic Voting Scheme |
| IA | Issuing Authority |
| ID | Identity |
| PK | Public Key |
| PV | Private Key |
| SD | Security Domain |
| SK | Session Key |
| TEE | Trusted Execution Environment |
| TPBV | Traditional Paper Based Voting |
| TUI | Trusted User Interface |
| TVA | Trusted Voting Application |
| UUID | Unique Universal Identifier |
| VC | Voter's Card |
| VT | Voting Terminal |
| REE | Rich Execution Environment |
| REV | Remote Electronic Voting |

# Chapter 1

# Introduction

For citizens of a nation to have faith in its democracy, they must be assured of the integrity of the ballots, secrecy of the ballots and reliability of the outcome of the elections after ballots have been counted. In most electronic voting schemes, both in the literature and schemes deployed for binding elections, trust is placed on various entities in the voting process to act honestly, if these trust assumptions hold, then the security requirements of an electronic voting scheme is achieved.

However, these trust assumptions may not always hold in practical deployment due to threats that exists in actual binding elections that may not have been considered as at the time of design. This breach in trust gives rise to the notion of an untrustworthy environment, which we define later in the thesis as a voting environment where voters and poll-site officials may be potentially corrupt, hence requiring extra security to mitigate against potentially malicious behaviour.

In many countries in the world, more so in developing economies, voter registration still remains a highly contested part of the electoral process. This is because of inadequate means of identifying legitimate voters due to poor documentation[1] thus creating an avenue for electoral fraud. Hence, eligibility verifiability is a very important security requirement in mitigating electoral fraud such as ballot stuffing, ineligible voting and

[1] https://www.cgdev.org/blog/finding-missing-millions-identity-and-sdgs

impersonation of legitimate voters. In many electronic voting schemes, voter authentication is done externally to the voting scheme. An example is the Voter Verifiable Scheme (vVote) [159] used in the State of Victoria in Australia which relied on poll-workers to authenticate voters and prevent ballot stuffing using human procedures and processes.

In this thesis, We discuss issues that motivates electoral fraud and identify some of the threats that exists in real world binding elections. Based on these issues we define two untrustworthy voting environments.

We do a security analysis of some electronic voting schemes based on our threat model and show why these schemes are not secure in an untrustworthy voting environment. Based on the existing infrastructure in our defined voting environment, We propose a novel generic mutual authentication scheme that addresses the issues of voter and ballot authentication whilst still preserving anonymity of the voter in an untrustworthy voting environment. Our proposed scheme is a front of a voting scheme which we then incorporate into existing e-voting techniques such as mix-nets and blind signature schemes. A formal verification of our proposed scheme using Scyther to show it satisfies the eligibility verifiability and other e-voting security requirements is done.

Finally, we do a threat analysis of an untrustworthy mobile voting environment and propose a novel voting scheme using group signatures and Trusted Execution Environment of a mobile device.

## 1.1 Motivation for Research

Voting has existed in communities for a long time and it is the process the populace use in expressing their political choices in a bid to elect candidates into office. In traditional paper based schemes (TPBV), votes are cast using paper ballots usually using the Australian Ballot System. In the Australian Ballot System, the names of candidates and parties are printed on a paper ballot. Voters are handed a blank ballot

after authentication, voters can either tick a box next to their preferred candidate, write in a candidate or thumb print on their preferred candidate using an Ink, after which the completed ballot is dropped in a ballot box. The ballots are then collated and counted manually to get a result.

Electronic voting on the other hand is voting using some form of electronic means. Electronic voting promises to improve on on shortcomings of TPBV by improving on speed, flexibility, security and accuracy of the voting process. However, most countries have not adopted electronic voting despite all its promises. This is mainly due to new threat realities that exists in an electronic voting environment that did not exist in traditional paper based schemes. Furthermore, achieving security properties such as ballot secrecy, individual verifiability, eligibility which is easily achievable in TPBV in certain environments, is not trivial to achieve in EVS.

Remote electronic voting (REV) which is voting over the internet in an unsupervised environment creates some extra threats that are easier to address in a supervised voting environment. As an example, in TPBV and supervised electronic voting, a voter goes into a private voting booth, casts a ballot with no one looking over his or shoulders. Where as in REV, it is difficult to prevent someone watching how a voter voted or voting on behalf of a voter. More importantly, if voters are not properly authenticated then the voting system cannot tell if an eligible voter or an ineligible voter has voted. If ballots are not authenticated, then the system cannot prevent ballot stuffing.

Usability of a voting scheme is important to consider when building a voting scheme intended for use in real world binding elections. Electronic voting schemes require complex cryptographic techniques to satisfy e-voting security requirements, this makes voting schemes complex and difficult to explain to the majority of voters that are non-technical. Furthermore, a simple system malfunction, cryptographic vulnerability or attack on an EVS can have major adverse effect on the outcome of the election as compared to TPBV elections. These issues and many more have slowed down the

adoption of REVS in advanced democracies [139] despite its positives.

Nevertheless, since verifiable electronic voting was introduced by Chaum in 1981, there has been massive progress made in the research and implementation of electronic voting schemes. Some countries have adopted electronic voting and used it in binding elections [159, 78]. In developing economies and hostile environments such as countries in Sub-Saharan Africa, Middle East, Asia etc. there is a clamour for the adoption of electronic voting systems. The main driver for the campaign is the promise that electronic Voting will mitigate some of the electoral fraud that currently affects the integrity of elections in these regions. Many countries in these regions struggle with proper documentation of citizens[2], which inevitably has an impact on the electoral register. Furthermore, lack of widely accepted legal forms of identification of voters, creates an opportunity for electoral fraud. These electoral frauds have been widely reported [40, 72] and it includes ballot stuffing, voter impersonation, illegitimate voting, under age voting and voter disenfranchisement.

Pre-election violence and violence on election day have also been widely reported [32, 71, 143, 54] in elections. This violence sometimes lead to loss of lives. This has created voter apathy, which has adversely affected voter turn out in elections in some countries. In the 2019 Nigerian elections, it was reported that only 34.75%[3] of eligible voters actually voted on election day, which is the second lowest participation rate in the history of elections in Africa[4]. Some of the reasons for the low turnout were linked to voter's not believing their votes would count; fear for voter's safety due to election violence; over militarization in some regions and poor planning that lead to delay in delivering electoral materials[5] to polling stations.

Nevertheless, some of these countries are adopting some form of technology in the electoral process to improve voters confidence in the electoral process. The willing-

[2]https://www.cgdev.org/blog/finding-missing-millions-identity-and-sdgs
[3]https://www.idea.int/data-tools/continent-view/Africa/40?st=pre#rep
[4]https://www.icirnigeria.org/2019-election-nigeria-has-the-lowest-voter-turnout-in-africa/
[5]https://www.icirnigeria.org/voter-turnout-for-presidential-nass-election-is-lowest-in-recent-history-here-is-why/

ness by these governments in accepting the use of technology, can encourage further adoption of end to end electronic voting systems, which is believed will improve voter's participation and election integrity. Currently, Ghana, Nigeria, Congo, Cameroon, Kenya, Pakistan amongst other countries have adopted biometric technology for voter authentication [80, 137, 100] to reduce illegitimate voting and voter impersonation. This has brought some improvement but there is still a lot of work to be done.

Smartcard technology for electronic identification is being adopted by many countries in the world. Both in developing and advanced economies, citizens are being issued Electronic ID cards (eID). Some of these eIDs have biometric verification and cryptographic capabilities which is leveraged on to grant citizens access to government resources and payment solutions. In Estonia [78], the eID issued to citizens is used for voting in binding elections. In Nigeria [74], the biometric capability of the eID issued to citizens is used to authenticate voters during elections before voters are allowed to cast ballots.

Electronic voting schemes in the literature can either be generic and suitable for many environments or designed specifically for a particular environment. When these schemes are being designed certain trust assumptions are made about the voting environment. In real world deployment some of the trust assumptions about the voting environment, the technology used and cryptographic techniques that underpins the protocol may not hold which then makes the voting scheme vulnerable to attacks. This is may be because proper threat analysis have not been done, attacker's motivation, socio-economic issues etc. may not have been considered thoroughly during the design or deployment of the voting system. Also, the schemes may not be End-to-End verifiable.

End-to-End verifiable voting schemes are meant to guarantee voters that their votes have been cast-as-intended, recorded-as-cast and any interested parties can verify that ballots have been counted-as-recorded. However, in some end-to-end voting

schemes [159], eligibility verifiability is done external to the rest of the protocol, this creates vulnerabilities that can be exploited to carry out further attacks to compromise the integrity of the election if deployed in the instance of real world untrustworthy environment defined in this thesis as **Untrustworthy Environment I** . We consider this possibility later in the thesis and carry out security analysis on existing voting schemes to show how this can be exploited.

Mobile devices have progressed over the years from simple mobile phones used to send messages and receive phone calls to full computing devices. With billions of mobile devices used around the world[35], a mobile device is an attractive option for electronic voting schemes in a hostile voting environment, where the safety of voters is a concern. On the other hand, the internet was not designed with security in mind, so naturally, connected mobile devices inherit some of the inherent security weaknesses that comes with the internet. Moreover, malwares have become more prevalent in mobile devices and some of the common attack vectors includes browsing suspicious websites, phishing attacks and download of malicious applications on devices. This makes mobile devices an untrustworthy environment, which leads us to **Untrustworthy Environment II** which we mentioned earlier and define further in Chapter 6.

Many mobile devices in use today have a Trusted Execution Environment within it that provides security for applications even in the face of an adversary. Considering some of the threats to free and fair elections, such as ballot box snatching, voter intimidation, pre-election violence, election result manipulation etc. that exists in a hostile voting environment (**Untrustworthy Environment II**), voting using a mobile device will be pivotal in addressing these security challenges whilst still improving voter participation and flexibility.

Our main motivation for this thesis, is to build electronic voting schemes suitable for elections in untrustworthy voting environments that mitigate electoral fraud that mainly stems from inadequate means of providing eligibility verifiability in elections.

Our proposed schemes will be built on existing infrastructures, technologies and procedures that governments and citizens of these countries are familiar with and already in use in some elections. These technologies include biometric authentication, multi application electronic identification cards and TEE enabled mobile devices.

## 1.2 Research Question

**RQ-1** Is the assumption that pollsite officials are trustworthy enough to prevent voter impersonation, ineligible voting and ballot stuffing reasonable?

**RQ-2** How can voter impersonation, ineligible voting and ballot stuffing be prevented in an untrustworthy supervised voting environment.

**RQ-3** How can trust be established in an untrustworthy mobile device to run a voting application.

**RQ-4** How can voter anonymity and eligibility verifiability be provided at the same time in an untrustworthy mobile device

## 1.3 Contributions

Using existing electronic voting literature and reports on electoral fraud in real world election, we defined two untrustworthy voting environments. **Untrustworthy Environment I**, is a supervised voting environment and **Untrustworthy Environment II**, is an unsecure mobile device in a hostile voting environment. Based on the threats that exists in these environment, an analysis of Estonian iVoting scheme and Australian Prêt-à-Voter scheme is done to show vulnerabilities that can be exploited. We identify that trust placed on electoral officials should be moved to technical security to prevent ballot stuffing, voter impersonation and voting by unregistered citizens in the untrustworthy environments defined. The main contributions to this thesis are listed below:

1. Using existing literature and reports on binding elections, attacker's motivation is highlighted, a threat model is built and two untrustworthy voting environments are defined.

2. Using existing infrastructures in **Untrustworthy Environment 1** such as Multi Application Smartcards and Biometric Technology, we propose a generic mutual entity authentication protocol to prevent ballot stuffing, voter impersonation and voting by unregistered citizens. The proposed scheme is incorporated into a mixnet scheme, a security analysis is done that shows it satisfies the security requirements of electronic voting.

3. Using the capabilities of a Trusted Execution Environment and anonymous group signature, a novel electronic voting scheme on a mobile device is proposed. This scheme scheme mitigates the vulnerabilities of an untrusworthy mobile device and can be incorporated into an existing electronic voting scheme to add eligibility verifiability.

4. Trust is moved from poll workers in an untrustworthy supervised voting environment to a tamper resistant smartcard and authentication using anonymous group signature is incorporated into an existing Prêt-à-Voter voting scheme to provide eligible verifiability

## 1.4 Thesis Outline

- **Chapter 2:** We present a background of the research done on electronic voting schemes over the years. We discuss some of the main cryptographic algorithms used in providing privacy that most electronic voting schemes are built on. We discuss the security requirements of electronic voting schemes that electronic voting schemes, including our voting schemes proposed in chapters 5, 6 and 7 aim to satisfy.

- **Chapter 3:** We set the scene for our Generic Voting Protocol on a Trusted Execution Environment proposed in Chapter 6 by presenting a technical background for Global Platform Trusted Execution Environment. We discuss the security features of a Trusted Execution Environment and key capabilities of a TEE technology which can be leveraged on to build a secure voting scheme in an unsecure mobile device.

- **Chapter 4:** In this chapter, we identify and discuss some of the the threats that exists in real world elections in developing economies which leads us to the notion of an untrustworthy environment. We present a case for the issues that motivate electoral fraud. We carry out a threat analysis and identify the capabilities of the attacker. We do an analysis of the iVoting Scheme in Estonia and vVote: Voter verifiable scheme in Australia using the threat model built from issues that exists in untrustworthy environments. We discuss some of the issues identified after analysing these voting schemes and present the case for technical security over human procedural security in certain critical aspects of an electronic voting scheme.

- **Chapter 5:** We carry out a threat analysis of an untrustworthy voting environment, which we define as **Untrustworthy Environment I** and discuss the attackers capability in this environment. We show the need for eligibility verifiability which is the main security requirement we try to satisfy in this environment. We then present our proposed mutual authentication scheme using smartcard technology, biometric technology and other cryptographic mechanisms to provide eligibility verifiability in a supervised voting scheme in an untrustworthy voting environment.

- **Chapter 6:** We discuss some the issues that exists in a hostile voting environment from reports of elections in countries. A mobile is an untrustworthy environment,

so we highlight reason why mobile devices are not secure. We then make a case for the use of mobile devices to address the security challenges in these hostile voting environments. Finally, we propose a remote electronic voting scheme based on group signatures and leveraging on security capabilities of the Trusted Execution Environment of a mobile device. $Securityanalysisoftheproposedvotingschemeanddiscussionsabou$

- In **Chapter 7**, voter authentication and ballot authentication techniques used in our previously proposed electronic voting scheme are integrated into a Prêt-à-Voter scheme. A security analysis of this new updated Prêt-à-Voter scheme and brief discussion afterwards concludes this chapter.

- **Chapter 8**: The thesis is concluded with a summary of the content of all the chapters and a discussion on how research objectives were met.Some further works for consideration are presented and a section on bibliography follows afterwards.

## 1.5 List of Publications

1. Voke Augoye and Allan Tomlinson. Mutual authentication in electronic voting schemes. In 16th Annual Conference on Privacy, Security and Trust, PST 2018,Belfast, Northern Ireland, Uk, August 28-30, 2018, pages 1—2, 2018.

2. Augoye, Voke and Tomlinson, Allan, "Analysis Of Electronic Voting Schemes In The Real World" (2018). UK Academy for Information Systems Conference Proceedings 2018. 14. https://aisel.aisnet.org/ukais2018/14

3. Voke Augoye, Electronic Voting: An Electronic Voting Scheme using the Secure Payment card System. Technical report RHUL-MA-2013-10 (Department of Mathematics, Royal Holloway, University of London, 2013), http://www.ma.rhul.ac.uk/tech.

4. TEE Mobile Voting Scheme- To be submitted

# Part I

# Background

# Chapter 2

# Background I:Electronic Voting Background

*This chapter provides a background on Electronic Voting. The literature on Electronic Voting is very large but we attempt to provide a journey of the progress in this field. The background presented in this chapter is based on previous background work done in my Msc thesis that has now been updated.*

## 2.1 Background on Electronic Voting

Voting has existed in communities for a long time and it's the process the populace use in expressing their political choices in a bid to elect their leaders. Gritzalis et al [87] expressed the fact that elections are very critical for the normal functioning of society and it serves as a means for citizens to express their opinions in granting power to selected officials. It also helps in building trust in the government and their support for democracy.

There are different ways and environments voting can be implemented. Traditional paper based schemes are conducted in polling stations under the supervision of electoral

officials and often monitored by party agents and observers. In traditional paper based voting schemes (TPBV), votes are cast using paper ballots usually using the Australian Ballot System. In the Australian Ballot System, the names of candidates and parties are printed on a paper ballot, voters tick their choices and drops it in a ballot box. Voting research has evolved over the years from purely manual paper based process to more electronic means. Nevertheless, most elections conducted in the real world are based on TPBV schemes mainly due to the new threat environments electronic voting creates.

Nevertheless, the use of electronic devices in voting is known as electronic voting [27]. Electronic voting began in the early 1960's with the use of punch cards, in the 1970's optical mark sense ballot (which converts paper ballots to electronic forms) and its application in voting was being explored. In the late 1990s, about 25% of voters in USA were making use of the optical mark sense voting technology [87].

The concept of Remote Electronic Voting using cryptography for verifiable elections was proposed by Chaum [43], since then there have been a lot of work done in this area [157, 49, 146, 31, 79]. Some voting systems still use a hybrid system which is a combination of a manual process and an electronic process. An example is elections in Nigeria [25] where voters are authenticated using biometrics technology before being allowed to cast ballots using paper based Australian ballot system. In Estonia, voters that have already cast votes remotely can go the polling centre to cast a paper vote which overrides the remote cast vote because priority is given to paper votes [48].

Concerns have been expressed in the steady reduction of voter turn out in election and a that has increased the call for online voting to improve participation [87]. Due to the rapid growth in the use of computers, mobile devices and advances in cryptography there is a serious push for electronic voting since a lot of people already have access to the internet [59]. The main driver for the push for electronic voting in developing economies in Sub-Saharan Africa and some parts of Asia, is due to exploitation of

TPBV schemes to carry out electoral fraud. This issues has been widely reported and we talk about it further in chapters 4, 5 and 6.

Electronic voting gives elections the much desired mobility which can improve election participation. Absentee ballot systems have been present for a while [85, 75], this gives voters that are out of their local precinct the ability to participate in elections. The idea behind absentee ballots is what electronic voting is based on loosely speaking. However, a lot of concerns have been raised over the years about the risks of using electronic voting systems considering all the possible threats they face [126, 112] such as privacy issues, voter authentication issues and ballot verification issues amongst others. For an electronic voting system to be deployed for binding elections, it must be sufficiently robust to be resistant to different kinds of attacks; It must not be to complex, so voters can understand and able to use it; Voters should be confident about the integrity of counted ballots because the integrity of a voting system is paramount to the integrity of any democratic system [112].

The Direct Recording Electronic (DRE) voting systems with an interface that is used in capturing votes directly, have been used in elections in the USA. After the discrepancies in the 2000 presidential elections, concerns have been raised about the security of the DRE system. The trust placed on the underlying system and lack of audit trail prevents it from satisfying the Verifiability of ballots cast. Neumann [133] says the DRE voting machines gives no assurance that ballots cast are properly tallied and processed since it has no guaranteed audit.

The same concern was expressed in the CALTECH MIT voting project [51] about the need for an effective and efficient audit trail; they proposed a system using audio which they called Voters Verified Audio Audit Transcript Trail (VVAATT). They also compared their system with the Voters Verified Paper Audit Trail (VVPAT) introduced by Mercuri et al [126] in 1992. Both systems have their shortcomings but they both provide a means through which a voter can verify that they have chosen the correct

candidate and an audit trail can be used to verify that there were no discrepancies or large case of electoral fraud.

Neumann [133] also expressed concerns about errors occurring frequently during elections mainly due to system operators rather than the programmers and the lack of assurance that intentional electoral frauds do not occur . All these concerns gave rise to verifiability which is discussed under security requirements for electronic voting in section 2.2

Despite the challenges and concerns with electronic voting, some some countries have used electronic voting for binding elections and citizens of many other countries are clamouring for electronic voting to improve the integrity of elections.

## 2.2 Electronic Voting Security Requirements

Credibility of elections can have very big impact impact on any society and its democracy. Citizens can lose trust in the system if there are any discrepancies or foul play in the electoral process, thus security is very important for an e-voting system.

Electoral fraud is a threat to electronic voting [19] and stability of democracy, so security is very important to prevent the realisation of these threats. Electronic voting is quite different from E-commerce so requires a much higher level of security than E-commerce. As an example anonymity which is a strong requirement for electronic voting might not be required in an E-commerce system. The type of election and voting environment an electronic voting scheme is deployed goes a long way in determining the security requirements they need to satisfy. For example, an electronic voting scheme needed to choose a student representatives in a university will not need to satisfy the same security requirements as a large scale general election in a country.

The environment an electronic voting scheme is designed can determine the security requirements it needs to satisfy. In a remote voting environment a voter can be monitored whilst voting but in a supervised voting environment, procedural controls should

prevent anyone from monitoring a voter whilst casting a ballot in a voting booth. Going through the electronic voting literature, we have listed below security requirements electronic voting protocols try to satisfy:

1. Privacy: This security property requires that voter's identity should not be linked to vote cast for example if a Voter Alice casts a vote XYZ, it should be impossible for an unauthorised 3rd party to link the vote XYZ to Alice. This means that the system shouldn't be able to reveal how the voter voted as defined in [63], thus voter's identity should remain anonymous [108]. Even if after the elections are concluded, the voter's privacy should be guaranteed [39].

2. Democracy: Any electronic voting protocol or system should be able to ensure that only eligible voters are allowed to vote and the protocol should also prevent the eligible voters from voting more than once [109], this property is defined in [63] as Eligibility and in [79] as un-reusability (i.e. a Voter cannot vote twice)

3. Receipt-freeness: This property ensures that a voter does not get any information that he can use to prove to a coercer that he voted in a certain way [63]. This property helps to prevent vote buying and vote selling by eligible voters. According to [31], this is the property that allows the electronic voting scheme meet the security of the secret ballot offered by tradition paper based schemes.

4. Fairness: If voters already have an idea of how people voted before they cast their votes, it may influence their decision. So this property ensures that all candidates are given a fair chance by preventing the release of any partial tally such that even counting officials have no clue about results [39] and voter's decisions are not influenced [20].

5. Accuracy: This property requires that all valid votes are counted correctly, invalid votes are not added and valid votes are not modified, removed or invalidated

from the finally tally. If any vote manipulation happens, it should be easily detected [39, 20]. This property is defined in [79] as Correctness.

6. Uncoercibility: This property ensures that a coercer cannot force a voter to get the value of his vote, or make the voter to cast votes in a particular way or for a particular candidate [39, 33]. Even voting authorities should not be able to derive the value of the vote.

Many electronic voting schemes make assumptions about the physical conditions of the voting environment in a bid to satisfy the security requirements listed above. Some of these assumptions include existence of a one way anonymous channel from authorities to voters [125] or an untappable channel [142]; Existence of trusted authorities [109]; Supervised voting booth by electoral officials [142]. These assumptions are pivotal in determining the security requirements necessary in electronic voting schemes.

### 2.2.1 Contradictory Security Properties of an E-voting

It is very difficult to satisfy all the security properties of an electronic voting scheme at the same time because quite a number of them are contradictory. This is known as the electronic voting problem [87, 39]. Privacy requires that a voter cannot be linked with the vote he casts (Ballot), while Verifiability requires that an observer should be able to verify the legitimacy of the voter and the integrity of the vote cast. Achieving both properties is especially difficult because it is hard to audit an election to ensure that every vote cast was by an eligible voter without compromising the privacy of the voter and his vote.

In the same light, Individual verifiability requires that voters can check that their votes were included in the final tally and they have not been tampered with. Individual verifiability is usually achieved by giving voters a receipt at the end of the election to confirm this. On the other hand, these receipts can then be used by coercers to

ensure voters voted in a certain way or by fraudulent voters in selling votes which is contradictory to the Receipt-freeness/Uncoeribility property which requires that voters get no proof they could show to a third party.

Although Efficiency is not a security property of an electronic voting scheme but achieving the other security properties i.e. accuracy, robustness, Universal Verifiability requires the use of cryptography that have high computational demands which may affect the usability and efficiency of an electronic voting scheme [87].

### 2.2.2 Verification and Auditability

Individual and universal verifiability are very important security requirements in electronic voting. The voting environment and assumptions made about the electronic scheme will determine how important verifiability is and how it can satisfied. Some voting schemes, voters are expected to trust the integrity of voting machines [48, 78, 11] to guarantee that their votes have been included in the final tally [47]. With the high rate of electoral fraud explicit trust cannot be placed on voting machine and authorities, this is the rationale behind schemes that try to provide voter's verifiability and auditability [17].

In [89] it was recommended that voters should have a physical record they can check to ensure that their votes are added to the final tally. Chaum proposed a voting scheme that ensures that a voter can confirm that their vote is included in the final tally [47]. Chaum's scheme [47] maintains ballot secrecy and provides high degree of transparency using high number of cryptographic techniques.

A scheme known has prêt a voter has also been proposed, this scheme aims to achieve assurance from the fact the election is auditable rather than placing trust on the system components or electoral officials [158]. The philosophy behind this scheme is end-to-end voter verifiable election where voters can verify that their votes are included in the final tally and auditors can audit every step of the voting process to detect any electoral fraud [158]. Chaum's visual cryptographic scheme [41] has inspired the prêt a

voter approach and several work has been done based on this approach [175, 157, 45]. In the prêt a voter scheme, a receipt is given to the voter which can be use in verifying their vote.This scheme still maintains the receipt-freeness, because the receipts given to voters are encrypted thus cannot be used in vote buying and selling. There are still other schemes which gives a code to the voter rather than an encrypted receipt as seen in [45] and compatible with the US Opscan voting system [156].

The scratch and vote scheme [17] uses paper based ballots and aims at minimising trust, by providing a scheme in which voters can participate in the audit process on election day before they cast their own votes and can also verify their vote has counted. In [38], a scheme was proposed which provides voters with incoercible voter's verifiable receipts to satisfy the verifiability property of an e-voting scheme. The authors claims the scheme is an improvement on older schemes based on mix-nets [42] which do not scale well and can only give voters a fixed level anonymity which their scheme improves on to give voters who do not trust the system ability to control their degree of anonymity beyond the level the system provides by default.

### 2.2.3 Eligibility Verifiability in Voting Schemes

Eligibility verifiability provides assurance that only legitimate voters are allowed to vote and cannot vote multiple times, this security requirement is also referred to as democracy and un-reusability as mentioned in Section 2.2. Eligibility verifiability ensures elections satisfy the "One Voter One Vote" concept.

Depending on the electronic voting scheme and the assumptions made, the need for voter's verification might be the most important security property. Traditional paper based schemes, some electronic voting schemes used in binding elections, and many of other e-voting schemes proposed in the literature, place trust on the electoral officials to authenticate voters. In countries with proper documentation and ID card schemes, authenticating voters might be quite straight forward. However, many countries strug-

gle with poor documentation [173] of citizens which impacts on the credibility of the electoral register. Thus relying solely on pollworkers to verify voters in these regions can lead to electoral fraud [106, 67]. It has been reported[1] that about 2.4 billion people around the world do not have widely recognised means of legal identification. With this statics in mind, achieving eligibility verifiability in elections in certain countries will be a very daunting task. This implies that voter authentication is very pivotal in preventing ineligible voting, ballot stuffing, carousel voting and voter impersonation.

Eligibility verifiability is made up of two parts, voter authentication and ballot authentication. Voter authentication should guarantee that eligible voters cannot be impersonated. While ballot authentication is meant to prevent ballot stuffing. Some voting schemes may out source voter authentication to poll workers and satisfy ballot authentication using cryptographic techniques like digital signatures. Other scheme rely on pollworkers to authenticate voters and still prevent ballot stuffing, an example is the (Prêt-à-Voter) scheme used in Australia [62].

Some other voting schemes do not focus too much on the voter registration phase because they assume registration authorities should be able to register eligible voters only. Based on this, voters can then authenticate to access voting applications and cast ballots using passwords. Passwords are not the most secure means of authenticating voters as they can be stolen or shared. If passwords are stolen, voters can be impersonated and if shared, then votes can easily be sold in remote voting schemes. Both scenarios are undesirable in a binding elections. Du Vote scheme[86], Belenois [56] and its variant Belenois RF [55] all rely on passwords to authenticate voters. While using passwords may be suitable for elections, in a high coercion environment, where voters and pollsite officials cannot be trusted not to breach eligibility verifiablity, stronger means of authenticating voters may be required. In Sections 4.2.1, we discuss socio-economic issues in certain countries and how this can motivate voters and pollworkers to participate in electoral fraud such as vote buying and selling.

[1] https://www.cgdev.org/blog/finding-missing-millions-identity-and-sdgs

In Du Vote Scheme [86], after voters have been registered they are issued a hardware token similar to used in payment solutions for generating random passcodes. During elections, after voters have authenticated to the election server using their usernames and passwords, a code sheet is displayed on the voters computer. The code sheet is a truncated probabilistic encryption of names of candidates. Voter inputs the displayed codes into the hardware token and a new passcode is generated, the order in which these codes are inputted into the token will depend on the voter's choice of candidate. In any case, the voting platform cannot tell the association between the plaintext votes and voter's choice. Security of this scheme is dependent on accurate registration of voters, because during this phase tokens are given to voters. The token has a secret parameter tied to the voter, the voting server knows this relationship, ballots are authenticated because the new passcodes generated requires the use of the secret parameter tied to the Voter. Some level of trust is placed on voters not to give out these tokens, if not vote buying and selling cannot be prevented.

In Belenois scheme [56, 55], after voter registration, each vote is provided a signature key and authenticates to the voting server using login and password. With encryption key and signature key, the voter encrypts and signs his ballot. This process is done by the voter's computer, which is assumed to be trusted. Since this is a remote voting scheme, the voter is more likely to be using a general purpose PC, so this trust assumption may not hold in real world deployment considering the different malwares that can infects PCs. Nevertheless, after the voter authenticates to the voting server using passwords, the ballot box re-randomizes the ballot and adapts voter's signature on the ballot before the ballot is published. The ballot is adapted in a way that the voter can still verify his signature on the ballot using his signature verification key. The voting server prevents ballot stuffing by discarding any ballots signed with the same signature key. In another scheme Selene[160], that issuers voters tracking numbers in to verify their votes have been recorded as cast whilst still able to defend against coercion, voters are also authenticated using passwords. This scheme prevents ballot stuffing by

using signatures on the ballot.

In summary, the appropriate means of authenticating voters will depend on the voting environment and the nature of the election. In the environments which we define later on in this thesis, we require a stronger level of voter authentication than passwords and a strong means of ballot authentication.

## 2.3 Overview of Electronic Voting Schemes

In this section, we do taxonomy of the some e-voting schemes discussed in literature and group them into 4 main models, which is The Mix-net models; The Blind signature model; The Homomorphic encryption model; The anonymous group signature model. Then we do a general discussion and analysis on schemes that have been proposed over the years based on these models.

### 2.3.1 Overview of E-Voting Scheme Based on Mix-nets

A mix-net is a cryptographic alternative to an anonymous channel [12]. In a mix-net used for election for example, messages which is the vote are sent from several senders to several receivers which would be the talliers via a third party (mix server) and an observer cannot. The Voters prepares his ballot appends a random string to the ballot (Message) and encrypts this with the public key of the tallier who is the intended recipient1 .

The voter now appends another random string R1 to the message alongside the identifier of the tallier, this identifier enables the mix-server know who the message is intended for. The voter now encrypts this message with the public key of the Mix-server this is as described in message 1 of the protocol run.

In 1981 Chaum introduced mix-nets [43] and each layer of a sent message from a sender i.e. Alice to a receiver i.e. Bob is decrypted by each mix-server along the way from sender to receiver and at the end an external observer cannot observe the rela-

tionship between any sender in particular and recipient. This message is first encrypted with the public key of each of the mixes [103]. This type of mix-net proposed by Chaum is a decryption type mix-net with simple RSA mixes. These types of mix introduced by Chaum are not very resilient to failure on like the reencryption mixes [102] which has greater resilience according to [38]. The scheme introduced by Jakobsson [102] eliminates the use of zero-knowledge proof making it more efficient than previous schemes based on mix-nets [101] and also eliminates the issue of encryption of the same plaintext resulting into similar cipher text that could be detected as seen in [101] according to the author.

There are also user centric mix-nets [14] which allow users manage their privacy requirements. In this mix-net [14] proposed, resilience is increased due to the collaboration in the exchange of ballot between the voters and third parties [38], although this protocol was generic but it can be applied to an e-voting scheme. At the end of the exchange of messages nobody observing can tell the relation between any particular voter and votes cast. In this scheme [14] a third party (electoral official) verifies the identity of the voter to ascertain his eligibility. The third party acts as go between the voter and the tallier, the tallier trusts the third party and believes that the eligibility of the voter and validity of the vote has been verified although the tallier and the third party (election officials) cannot link the transactions back to a specific voter [14].

After registration all the voters are given a unique token, they all simultaneously submit this unique token to the third party who now issues out another new unique token in such a way that it cannot tell which voter got which token. In this approach of mix-net the user has to pay more attention to the process and although you can achieve the anonymity and privacy property but it is not very practical because of the increased user involvement and cost by possibly sending larger amount of messages [14].

In [147] the authors proposed a scheme which improved on the Chaum's mixnet [43]. According to [147] their scheme improves on the message expansion issue Chaum's

mix-net scheme had because in Chaum's scheme [43] the number of Mixers increases in relation to the length of the message making it less efficient than their scheme in which the length of the message is irrelevant to the number of mixes used. In the second scheme proposed in [147], they claim they improve on Chaum's scheme which provides very little level of correctness (i.e. a mix-net should ensure that an output corresponds to the input) and doesn't satisfy the fairness property meaning that if one vote is disrupted the outcome of the election can be learnt before the final tally is announced [87]. Further analysis was done on the scheme proposed in [147] and according to [150] the scheme can be attacked and secrecy of votes in the election scheme can be compromised. They proposed a countermeasure but they however did not guarantee that modifications to the protocol would make the channels or corresponding election protocol secure.

Abstractly a mix-net should achieve these 3 goals: A mix-net should ensure that the output corresponds to the input (the correctness property); an observer should not be able to link an input element to a given output element this property is known as privacy; a mix-net should be robust i.e. provide a proof that it has operated correctly which can be verified by all parties [103]. The scheme proposed in [103] aims at making mix-net robust by revealing a relation between the input and output which is selected pseudo-randomly by each mix-server as evidence of correctness in its operation. The process used in this scheme is known as "Randomised Partial Checking" [103]. According to [103] privacy is not dependent on a single server being honest like traditional mix-net schemes [43] rather it's a global property since every server reveals a portion of the relation between the input/output and even with corrupt mix-servers there is no way of connecting an input with a particular output. In 2001 Neff [132] proposed an efficient verifiable mixing technique that can be used to achieve universally verifiable elections. Voter's credentials are mixed before the election commences rather than mixing encrypted votes (cipher texts) after the vote collection centre has received the ballots.

In [47] Chaum proposed a scheme for electronic voting where voters get encrypted receipts to verify their votes and the tellers ensure there is no link between the encrypted version and decrypted ballot receipts by performing anonymizing mixes. Ryan and Schneider later proposed another scheme [158] which uses re-encryption mixes in the anonymizing tabulation phase instead of decryption mixes this has an advantage over the RSA decryption mix used in his earlier schemes by Chaum [158, 41] because its more tolerant to failure of any of the mix tellers and enables full independent rerun of the mixes and audit if necessary.

Mayasuki [12] proposed a robust e-voting scheme based on mix-net that is universally verifiable where the amount of mix-servers does not determine the amount of work done by the verifier i.e. the work done by a verifier is not dependent on the number of mix-servers. There have also been other literature based on mix-nets [168] and other literature attacking mix-nets to compromise the privacy of votes and robustness of the electronic voting system like the attacks shown in [171] which attacked the scheme proposed in [84]

### 2.3.2 Overview of E-Voting Schemes Based on Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows specific type of operation to be carried out on a ciphertext to obtain an encrypted result which is the ciphertext of the result of operations performed on the plaintext. For example one party could add two encrypted numbers and then another party i.e. voting authority that is in charge of vote tallying could decrypt the results without either of the parties being able to find the value of the individual numbers. With homomorphic encryption there is an operation defined on the message space and an operation defined on the cipher space such that the product of the encryption of any two votes is the sum of the votes [38]

In [31] Benaloh and Tuinstra proposed a scheme based on homomorphic property of a probabilistic encryption method (i.e El-Gamal) that provides the first verifiable

secret-ballot election protocol that prevents vote selling and coercion. They assumed the existence of a voting booth which should help prevent coercion and the fact that voters are not given a receipt would prevent vote selling. They also proposed two protocols in this scheme one is a single authority voting protocol which does not achieve the secrecy of votes and the second one which achieves vote secrecy, is a multi-authority scheme [31]. Both protocols use homomorphic encryption.

Martin hirt and Kazue sako [94] shows that the claims by Benaloh et al that their scheme is the first receipt free scheme [31] is actually not the case because it doesn't achieve receipt-freeness. They proposed a practical receipt free-voting scheme [94] based on homormorphic encryption with additional assumptions about the properties of the encryption function such as the decryption must be verifiable, the encryption must be infeasible to decrypt if the authorities are less than a certain number etc. This scheme proposed by them also takes advantage of efficiency of the protocol proposed by Cramer et al [58].

Cramer et al. [58] proposed a scheme based on homomorphic encryption and its special properties to guarantee privacy, Universal verifiability, and robustness. This scheme uses a variant of El-gamal encryption and it is part of the security of the scheme because of the computational difficulty in solving the discrete logarithm problem in El-gamal. According to the authors their multi-authority scheme reduces the task of the voters to the bare minimum [58]. The scheme [58] achieves universal verifiability i.e. any observer can verify the final tally due to the homomorphic property of the encryption method used. The scheme also achieves privacy of votes and robustness (i.e. failure of authorities can be tolerated) by the use of threshold decryption techniques whereby the final tallying process is shared among several authorities [87].

According to [58] the communication complexity both for the individual voters and authorities is minimal making performance of the scheme optimal. However if the number of the candidates is large then it would have a relatively high computational

complexity for this scheme based on El-gamal [87]. Another downside of the scheme and other schemes based on homomorphic encryption is the limitation of the votes to YES/NO value which reduces flexibility [87] and hence makes it not very practical for large scale elections with multiple candidates or choices.

In [149] the authors proposed a new voting scheme based on multiplicative homomorphism where the votes are recovered by decrypting the product of the votes on like the other schemes [94, 31] that are based on additive homomorphism where decryption is done on the sum of the votes. According to the authors, this scheme provides strong privacy, Universal verifiability and is more efficient than previous schemes based on additive homomorphism [149].

In [88] Groth investigated four types of e-voting schemes namely: Borda Vote which is a preference vote where the best candidate receives L votes and the second L-1 votes etc; Approval vote which is any number of L candidates; Limited vote (N out of L candidates where N is the number of votes the voter can cast); Divisible vote where a huge number of vote is distributed among the candidates. They also presented some efficient non-interactive zero-knowledge (NIZK) arguments based on homomorphic integer commitment. According to Groth [88], homomorphic threshold voting improves the efficiency of both Borda and Approval voting.

In [92] the authors presented a scheme that achieves receipt freeness for the Groth's e-voting scheme since the Groth's scheme does not achieve receipt-freeness due to the ability of a voter to construct a receipt (which can be used for vote selling) by exploiting the randomness she chooses in encryption or commitment. A lot of other schemes have been proposed based on homomorphic encryption, further details about them could be found in [110, 169, 148].

However concerns have been raised over schemes based on homomorphic encryption. In [149] it was expressed that mixing votes are said to be more efficient than homomorphic voting in elections where there are multiple choices and candidates be-

cause homomorphic voting requires each vote to be verified if not the validity of the tallying stage cannot be guaranteed hence it is restricted to YES/NO voting a similar view was also expressed in [87].

In [18] the authors compared two schemes using homormorphic encryption and mix networks in other to achieve preferential voting. The authors [18] expressed that as the number of candidates L increases then the preferential voting system is inherently larger than the 1-out-of L (where 1 candidate is chosen out of M candidates) voting system. Hence voting system using mix-networks are more efficient because the number of candidates do not adversely affect the computational complexity on like voting systems with a form of homomorphic encryption which tend to be inefficient or not practical [18].

### 2.3.3 Overview of Electronic Voting Schemes Based on Blind Signatures

The concept of blind signature was introduced by Chaum in his paper "Blind Signatures for Untraceable Payment" [44] as a form of digital signature in which the message is authenticated without knowing the content of the message [38]. The signer of the message cannot derive the correspondence between signing process and the signature which is later publicly available hence making this type of signature unlinkable. In electronic voting, the voter obtains a token which is a blindly signed token that only the voter knows, then the voter sends this token along with his vote to the appropriate electoral official [125].

Fujioka et al. proposed an electronic voting scheme [79](FOO Scheme) based on blind signature for a practical large scale election that ensures privacy of the voters and realizes voting fairness [142]. Other schemes have been proposed that are related to the work proposed in [142], Cranor et al. proposed a scheme called SENSUS [59], this is a Security Conscious electronic polling system for the internet, suitable for small scale elections but with minor modifications it can be used in large scale elections according to the authors [59]. SENSUS uses blind signature in a bid to provide privacy of vot-

ers. Both the SENSUS [59] and Fujioka's scheme [142] assume there is an anonymous communication between voters and election authorities. Both schemes also consist of a Voter, Registrar, Validator (administrator) and Tallier (counter). SENSUS has an extra central facility called a Pollster who acts as a voter's agent and performs all functions on behalf of the voters such as cryptographic functions and transfer of data functions [59, 109]. SENSUS doesn't prevent vote selling because a voter can prove he voted in a certain way, SENSUS satisfies individual verifiability but not universal verifiability.

In 1998 Mu et al. proposed two secure electronic voting schemes [129] to conduct elections over the internet based on El-gamal digital signature. This scheme ensures privacy is maintained and it prevents double voting [129]. The scheme uses a blind signature between the Voter and Authentication Server (AS) so that $AS$ does not have any information on voting tickets and other parameters to be used in future elections [129].

In 2003, the authors of [117] showed that the scheme proposed in [129] has some security flaws and that double voting which the authors believed their scheme [129] prevents is actually not the case because some voters can double vote without being detected. They then proposed a modification of the protocol used in the scheme [129], to prevent the double voting flaw and satisfy other security requirements like anonymity of voters, verifiability and correctness [117]. However, in [96] the authors show that the improvement on the scheme proposed in [129] by the Lin et al [117] allows authorities to break the voter's anonymity and hence compromising voter's privacy because the authorities can identify the owners of the cast tickets. They now proposed a new scheme to solve this issue [96].

In another scheme [155] based on the initial scheme proposed in [129], the authors looked at all the works done in [117] and [96] and according to them there is a high probability that voters would have difficulties signing voting contents. In their

scheme [155] they replaced El-gamal digital signature which was used in all the previous schemes [129, 117, 96] with Digital Signature Algorithm (DSA) and RSA to generate blind signatures, this they believed would solve the aforementioned issue. Another scheme based on the FOO scheme was proposed in [76] also based on blind signatures and it uses the already existing GSM infrastructure, taking advantage of the authentication method in GSM to propose an electronic voting scheme which gives efficient, transparent and mobile authentication of voters without compromising their privacy.

In 2008 the authors of [73] proposed an electronic voting protocol based on a dual randomized blind signature where voters get multiple receipts as a mean to provide individual verifiability while preventing coercion of or vote selling by voters because the coercer cannot tell which of the receipt is the actual vote. According to the authors their 34 multi-receipt concept is not theoretically perfect but it is suitable for a practical election, especially in cases where vote selling and buying in the elections are minimal.

Mohanty et al [128] also used blind signature for authentication and XOR operations to generate votes in their multi-authority electronic voting protocol that they say is suitable for large scale elections and in 2011 another blind signature scheme with its electronic voting protocol based on elliptic curve was also designed more details about this scheme could be found in [153].

### 2.3.4 Overview of Anonymous Group Signature Scheme

In anonymous group signature schemes, a group member is able to sign a message on behalf of a group anonymously, this scheme was introduced by Chaum and Van Heyst [46]. To verify a group member hasn't signed multiple messages, involves the group manager opening all the messages to identify members that have signed multiple times. This concept introduced by Chaum and Van Heyst [46] revealed the identities of all group members and it lead to the list signature scheme proposed by canard et al [36]. In the list signature scheme [36], the concept of linkability was proposed, with a linking tag, any party can publicly identify multiple messages signed by the same

group member. This makes list signature schemes attractive for electronic voting.

The ISO/IEC 20008 standard [3, 4] has standardised an anonymous group signature scheme with linking tags based on the link signature scheme proposed by Canard et al [36]. The voting schemes proposed in Chapters 6 and 7 are based on this standard.

Another anonymous signature concept was proposed by Rivest et al [154], this is called a ring signature scheme. In this scheme, every group member has a unique signature key and public key. To sign a message, the true signer takes as input the message, the true signer signature key, the true signer's public key and public keys of all other members in the group. The ring signature scheme proposed by Rivest et al [154], does not have a group manager and multiple messages signed by a true signer cannot be linked. Dodis et al [65] proposed a ring signature scheme that involves a group manager with the ability to open messages to identify group members that have signed multiple messages. Dodis et al scheme [65] would not be suitable for electronic voting since ballots cannot be linked without revealing the identity of every member in the group and messages they have signed.

Democratic Group Signature (DGS) schemes was introduce by Manuli et al [121, 120], a verifier outside of a group cannot distinguish which group member signed the message, preserving the anonymity of group members to outsiders. However, anonymity of a signer is not guaranteed within the group as individual members of a group can trace and reveal the identity of the signer of a message. This makes the DGS scheme vulnerable to insider attacks and unsuitable for electronic voting schemes.

Xiangxue Li et al [116] proposed a DGS scheme with collective traceability that extends on Manuli's DGS scheme [121]. This scheme has no group manager and in the case of a dispute all the members of the group need to collectively cooperate to generate a secret to reveal the identity of the signer. However in an electronic voting context, this scheme would be unsuitable as it would require all voters even abstaining voters to cooperate to reveal the identity of a cheating voter. Electronic voting schemes

requires voters to vote and go, this cannot be achieved if voters are expected to take up this collective traceability responsibility.

Moreover, without a dispute which may arise from more ballots being signed than the total number of voters in a group, then there might be no need for this collective traceability. This would imply that multiple ballots signed by a group member may not give rise to suspicions if the total number of signed ballots is still less than the total number of members in the group. Besides revealing the identity of a voter is not be a desirable security goal of a voting scheme because that breaks voter's privacy. Identifying multiple votes cast by an eligible voter without knowing the identity of the voter is sufficient for electronic voting, further making the DGS schemes with individual [120, 121] and collective traceability [116] unsuitable for electronic voting.

Pan et al. proposed a voting scheme based on ring signature called RE-NOTE [146] this scheme is an improvement on an earlier scheme they proposed called E-NOTE [145]. The ballot distribution phase of E-NOTE [145] involves issuing a voter a blank ballot after the voter presents a certificate issued to the voter by a Voter Registration Authority. RE-NOTE [146] scheme added ring signatures to the ballot distribution process. The ring signature scheme in this protocol does not offer ballot linkability, to achieve this the scheme introduced a watch dog device that records every process of the scheme. A blank Ballot is distributed to a voter, if the ring signature over a randomly generated number sent to the Ballot Distribution Centre by the voter is valid, voters are only given one ballot to prevent double voting. Since this device records every process during the election, it has to be completely trusted not to reveal the link between the voter's identity and ballot which is possible if all the processes are pieced together. The use of the device creates a single point that could have been avoided if the ring signature scheme offered linkability.

Another group signature voting scheme was proposed by Malina et al. [118] using the group signature scheme from [119], this scheme prevents ballot stuffing by issuing

every voter a Unique Token generated from the Voter ID rather than a signature linking tag. The scheme has a Group Manager that generates the group signature keys and voter ID for the Voter. Another Authority called the Polling Station (PS) generates a Unique Voter Token from the Voter ID, PS also generates election encryption keys and tallies final ballot counts at the end of the election. During the tallying phase the PS knows the voter's ID, Voter's Token and Voter's candidate choice after it decrypts the encrypted and group signed message from Voter that contains the Unique Token and Filled Ballot. Even though the link between Voter's ID and Voter's True Identity is only known by the Group Manager, the PS has formed a link between the Voter ID, Unique Token and Plain ballot, this already breaks the anonymity of the Voter ID. Malina et al [118] argue that privacy is still maintained since PS does not know the relationship between Voter ID and True ID. The Group Manager needs to be trusted not to cooperate with PS to reveal the Voter's True ID, if not the whole scheme would be broken. This issue exists, because the group signature protocol proposed does not offer ballot linkability but depends on trust spread across multiple authorities.

## 2.4 Electronic Voting Scheme in the Real World

Electronic voting can help improve the flexibility and mobility of elections. Electronic voting can also help reduce electoral fraud widely reported in elections in some countries. Many countries still use paper based schemes while some other countries adopt some form of electronic processes at some point within the voting process. As an example some countries [80] deploy bioemtric technology for voter authentication but the rest of the process is still based on manual Australian ballots.

Nevertheless, electronic voting have been deployed in real world elections. A scheme based on Prêt-à-Voter was used in elections in Australia, we discuss this scheme further in chapters 4 and 7. Estonia [93] and Switzerland [162] have also used internet voting schemes in several elections. In 2012 [53], Expatriates used internet voting for

parliamentary elections in France

The city of Takoma, Maryland became the first to use an end-to-end verifiable voting scheme called Scantegrity II [37] in a binding government election in 2009, it was also used in 2011 elections.

In Norway, Approximately 160,000 voters have voted using remote internet voting in two pilot elections [163] conducted in 2011 and 2013. In the Norwegian voting scheme, the voter is sent a code sheet that contains codes of the various candidates prior to the election. The voter logs in using a two factor authentication mechanism called MinID, which is a well-known mechanism in Norway. After the voter is verified, the voter inputs a code into an applet on his computer, the code should correspond to the code for the party of his choice contained in the poll card received by the voter prior to the election. The poll card is sent via the postal service. The applet then encrypts the code and forwards it to the central voting system, which forwards a return code via an SMS back to the voter. The voter compares the return code with the code corresponding to the party of his choice on the poll card, if both codes are the same then his vote has been recorded-as-cast [57, 163]. The Norwegian internet voting scheme assumes server infrastructures are not under the control of adversary and SMS post-channel cannot be intercepted, interrupted, or manipulated by the adversary. However this scheme was analysed and some attacks carried out on the SMS channel, blocking and fabricating SMS sent to the voter deceiving the voter into believing their votes have been cast-as-intended [111]. Other attacks were carried out on the MinID authentication systems

Helios [15] is a web based open audit voting system and it has been deployed for organisational elections [16, 91]. Anyone can setup an election using Helios, invite voters to cast secret ballot, tally the ballot and verify the integrity of the election. Helios provides verifiability by allowing voters audit ballots to confirm correct encryption, before discarding the audited ballots and casting a final ballot not audited. Helios also uses reencryption mix-net to provide privacy of the voting scheme.

# Chapter 3

# Background II: Technical Background

*GlobalPlatform*[1] *is a non-profit industry association driven by approximately 100 member companies. They develop specifications enabling digital services and devices to be trusted and securely managed throughout their lifecycle. The GlobalPlatform have developed specifications for a Trusted Execution Environment in a Mobile Device. A Trusted Execution Environment, is a isolated secure environment that alongs along side an unsecure Rich Execution Environment where an application can execute security sensitive operations. This chapter discusses some of the key parts of a TEE architecture needed for the Electronic Voting Protocol in Chapter 6*

## 3.1 Introduction

With about 1.56 billion smart phones shipped yearly as at 2018[2], most service providers from banking applications, to eCommerce websites and government agencies have applications running on smartphone. The ability to securely manage applications on

[1] https://globalplatform.org/why-globalplatform/overview/
[2] https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/

smartphones, carrying out security sensitivity operations such as encryption of data, authentication of users, ensuring integrity of data makes TEE a very attractive technology that would be suitable for running electronic voting schemes in an otherwise unsecure mobile environment.

In this chapter we present some background on GlobalPlatform Trusted Execution Environment (TEE) as a technical mechanism to provide trust within an untrustworthy mobile device required for our voting scheme proposed in chapter 6. There are other TEE technologies (ARM TrustZone [23], Intel SGX [124]) but GlobalPlatform TEE is considered in this thesis because it is more generic.

The security features of the TEE Architecture is discussed in this chapter. Also described is the Security architecture of a TEE; Key Entities in a TEE and communication between Client Application in the Rich Execution Environment (REE) and the Trusted Application (TA) in the TEE.

## 3.2 Trusted Execution Environment

A Trusted Execution Environment, is a secure isolated environment on the main processor on a mobile device, it runs in parallel to a Rich Execution Environment which contains one or more operating systems [83, 5]. A Trusted Execution Environment provides confidentiality and integrity guarantees for data and applications loaded in this environment making it attractive for service providers such as financial institution, government etc. to take advantage of the security capabilities of the TEE.

The GlobalPlatform and Trusted Computing Base have various specification [5, 7, 83, 10] of how to a TEE can be implemented on a mobile device and User centric approaches to TEE according to the specific needs for the service provider. Currently Samsung PAY uses ARM TrustZone [166] technology for trusted execution on android smartphones but this is an Issuer Centric TEE application but the concept is the same for an User Centric Model

The TEE technology would form a building block for the electronic voting protocol proposed in Chapter 6

### 3.2.1 Security Features of TEE

In this section, some of the security features of a Trusted execution Environment as described in various GPD standards [5, 2, 83] is described

**Isolation from the Rich OS** : Due to the resource limitation in mobile devices, a TEE has a Rich Operating System which has the full features of the application whilst security critical aspect of the application is ran in the TEE environment as a Trusted Application. The TEE must provide an isolation between the unsecure Rich OS in the Rich Execution Environment and Trusted Applications, its related data within the Trusted Execution Environment

**Isolation from other TAs and TEEs** : The GPD specification [83] for TEE, provides for multiple TAs within a TEE and Multiples TEEs within the device. Trusted Applications are isolated from other TAs within the TEE, and from the TEE itself. Also, TEEs are Isolated from other TEEs. This implies that every TA and TEE manager their own resources that cannot be accessed by other TAs and TEEs within the device. If a relying TA requires a service from another TA, the relying TA must have the permission to request for the service and that request would be authorised before services are granted to the relying TA. This is relevant because security sensitive operations for a particular Client Application can be spread across multiple TAs, and TAs must TAs must communicate securely to share services

**Application management control** : Any modification of the TA and the TEE can only be performed by an authorised entity and this entity is authenticated before any modification can be made. Some security domains might have the authority to carry out operations on other security domains and TA applications. For

example some Security Domain can create other Security Domains, to do this, the SD needs to have the right permission.

**Identification and binding** : The boot process is bound to the System-on-Chip (SoC), enforcing authenticity and integrity of TEE firmware and TAs.

**Trusted storage** : The TEE offer secure storage of data and keys.The Data stored on a TEE cannot be accessed, copied or altered by an unauthorized internal or external entity because the storage is bound to a specific TEE on a particular device. Hence the TEE offer anti-cloning protection.

**Trusted access to peripherals** : TEE offers Trusted Applications the necessary APIs to access to Trusted Peripherals such as the touch screens, key boards, biometric sensors and other peripherals, under the control of the TEE.

**Secure Pathway:** The TEE offers a Trusted User Interface (Trusted UI) that ensures information which appears on a device screen comes from a Trusted TA and any input from the user is isolated from other applications. This is a particularly useful security feature which would be important when filling out ballots in the electronic voting proposed in section 6.

**Remote Attestation:** Remote attestation is the assurance that a particular piece of software has not been modified without adequate authorisation. This means that the TEE can confirm the integrity of a software. This is an important feature of a TEE and one which is very attractive for an electronic voting scheme in an untrustworthy mobile device.

**Cryptographic Capabilities:** A TEE can generate random numbers, cryptographic keys and carry out cryptographic operations such as encryption, decryption, digital signatures and hash functions.

### 3.2.2 Key Entities of a Trusted Execution Environment

This section discusses some entities of a Trusted Execution Environment Architecture as defined in the GPD Standard [83, 5] and relevant for the voting protocol proposed in Chapter 6

1. Client Application (CA): This is an application running in the unsecure Rich Execution Environment outside of the Trusted Execution Environment on a mobile device. The client application makes use of the TEE Client API to access services provided by Trusted Applications located in the TEE.

2. Rich Execution Environment: This is an unsecure execution environment that consists of a Rich OS, device components.The client application runs in this environment. Anything that runs in the REE is considered as unsecure by the TEE

3. Rich OS: This is an Operating System that runs in the REE and provides a wider range of features than the Trusted OS running in the TEE. The Rich OS has a much higher functionality and performance than the Trusted OS, although it is considered as unsecure by the TEE.

4. Communication Agent: TEE and REE communicate through drivers. The REE and TEE communication agent is a Rich OS driver and Trusted OS driver respectively that enables such communication

5. Security Domain: This is an on-device representative of a Remote Authority . Security domains are used to manage the TEE, other Security Domains and Trusted Application depending on their privilege. Sections 6.5 shows how to create a security domain, generate keys and provision keys for a Trusted Application in the Electronic Voting Scheme proposed in Chapter 6

6. Service Provider: The owner or vendor of a combination of Client Application and Trusted Application software. In our scheme proposed in chapter 6, this is the Election Authority

7. TEE Client API: The Client Application running in the Rich Execution Environment communicates with an Instance of the Trusted Application in the Trusted Execution Environment using the TEE client API.

8. TEE Internal APIs: This is a general series of APIs that provide a common implementation for functionality often required by Trusted Applications. Examples includes the Trusted Storage API, Cryptographic Operations API, Peripheral APIs etc. Some the APIs is discussed in this section 3.3.1

9. Trusted Application (TA): This is an application running inside the Trusted Execution Environment that provides security related security related services to Client Applications running in the REE. It also provides security services to other Trusted Applications with the right authorisation inside the TEE. In our proposed scheme in Chapter 6, the Trusted Voting Application (which is a TA) provides security services such as ballot encryption and anonymous signature over the ballot for the Voting Application (which is a CA)

10. Trusted Execution Environment (TEE): As defined in section 3.2

11. Trusted Device Driver: Communication between Trusted Applications and TEE hardware is done secure through the Trusted Device Drivers resident in the TEE.

12. Trusted OS: A TEE has only one Trusted OS, this OS runs within the TEE and provides the Internal APIs within the TEE that Trusted Applications require to access Trusted Resources. It also provides a TEE client API through which A TEE can communicate with a REE.

13. Trusted storage: This is a secure storage within the TEE that is protected either by the hardware of the TEE, or cryptographically by keys held in the TEE. The TEE secure storage is bound to the device so it offers far more security than can be offered in the Rich Execution Environment

## 3.3 TEE Architecture

In this section the generic TEE hardware and software architecture [83] is discussed. In actual implementation, how these architectures look would be dependent of the Original Equipment Manufacturer (OEM) or the TEE Provider, figure 3.1 shows an architectural view of the hardware architecture with no specific implementation.

For the hardware architecture there are different possible implementations for embedding the TEE on a device described in the standard [83]. The TEE can be embedded on a device System on Chip (SoC) or mounted on the main device Printed Circuit Board (PCB). If the TEE is mounted on the main device PCB rather than on the SoC then it doesn't support a Trusted User Interface and is managed by an External Secure Element, this implementation would not be suitable for the voting scheme proposed in Chapter 6 because the scheme heavily relies on the Secure Display of ballots and Secure Input of candidate choice provided by the TUI.

Another architecture in the standard, the TEE is within a Secure Element embedded in the SoC, the Secure Element is a resource restricted area. Ownership and administering control over an Internal SE would bigger business consideration than a Trusted Execution Environment directly within a SoC. So the focus is more on the TEE within the SoC in this chapter shown in Figure 3.2.

In the third Implementation described in the standard [83] and shown in figure 3.2, the TEE is within the SoC and has access to External Memories. Both the REE and TEE share the ROM, Micro Processor, Crypto-accelerators, RAM, Peripherals etc.These components are either Trusted or Untrusted or they switch between states

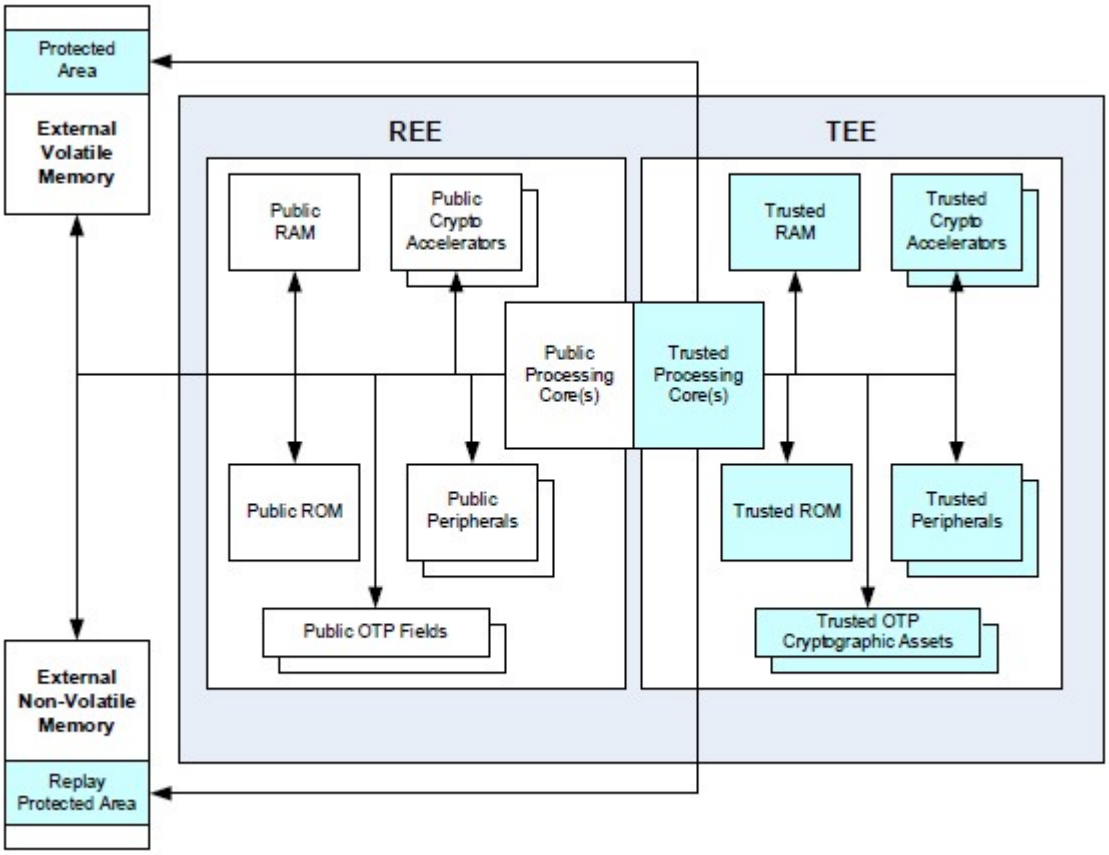


Figure 3.1: Hardware Architecture REE and TEE [83]

from an Untrusted State to a Trusted State [22].

TEE does not require permission to access resources shared with REE but controlled by REE. However, resources controlled by the TEE cannot be accessed by REE and other TEEs without specific permission granted.

The software architecture fig 3.3 is made up of the Rich Execution Environment (REE) and the Trusted Execution Environment (TEE) that runs alongside it. These 2 environments are isolated from each other from sharing resources and components using physical isolation, cryptographic isolation or logic based isolation methods. Nevertheless, REE can access trusted resources in TEE using APIs, for example Client

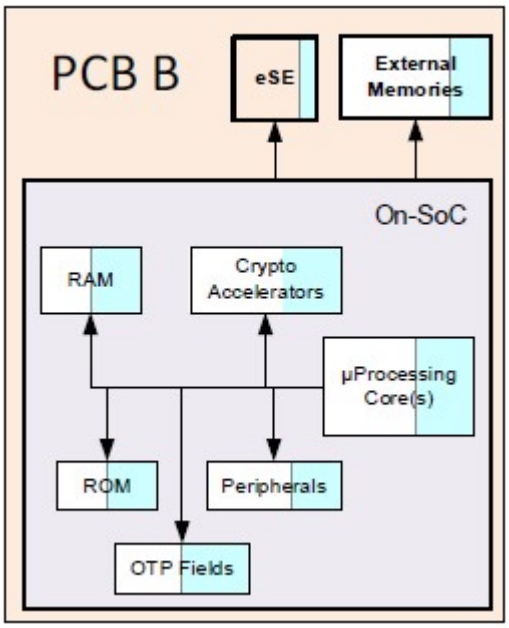


Figure 3.2: TEE Hardware Implementation System on Chip [83]

Applications that runs in the Rich OS within the REE can access the Trusted Application that run in TEE through a Client Internal API(see definitions 3.2.2).

When a sessions is created by CA with TA, CA connects to an instance of TA. The Physical memory address space of all the other TAs is separated from the physical memory address space for this instance of TA. As shown if figure 3.3, within TEE , there is a Trusted OS Component which provides a hosting code and the Trusted Application runs on top of this code. The Trusted OS component is made up of the Trusted Core frame work and the Trusted Device Drivers. The Trusted core frame work provides OS functionality to the Trusted Application while communication interface to Trusted Peripherals is provided by the Trusted Device Drivers.

REE communicates with TEE through a collaboration between a TEE Communication agent (Trusted OS Component) and REE communication agent (Rich OS component)

A typical Client Application, will establish a session with a Trusted Application in the TEE by communication with the TEE using the TEE client API. After which a

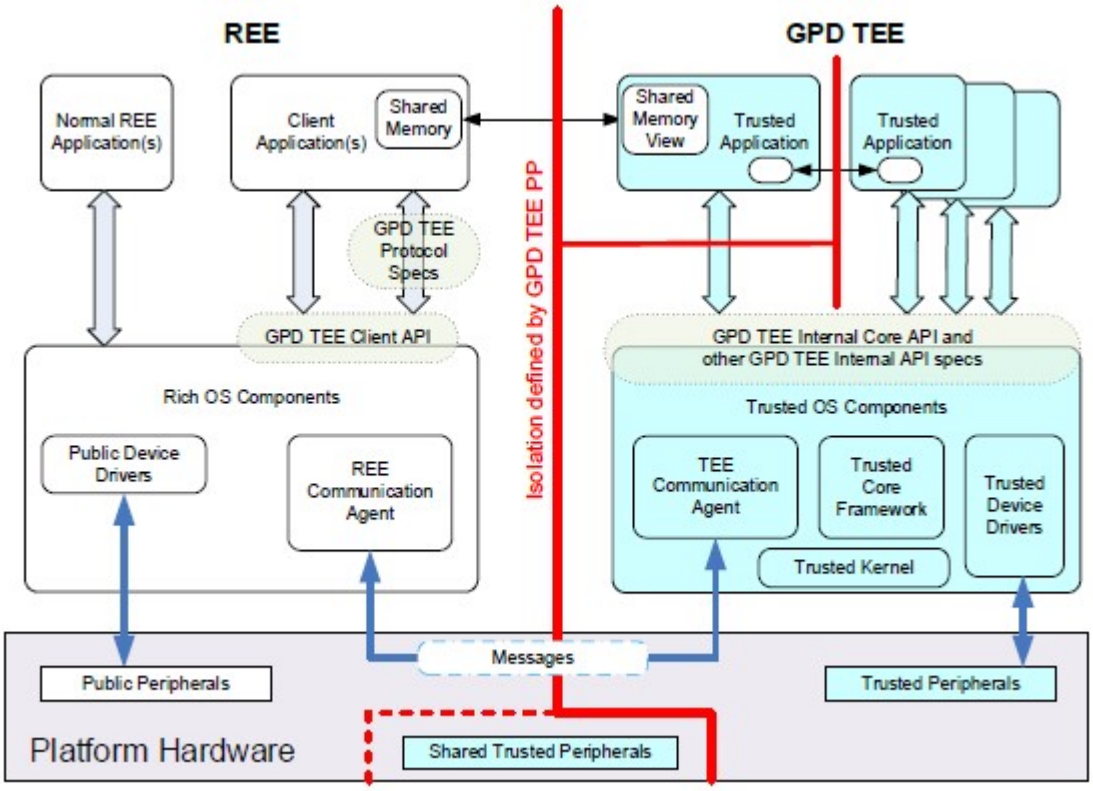


Figure 3.3: TEE Software Architecture [83]

shared memory is set up between CA and TA, with the shared memory both parties can exchange large amount of data efficiently. CA then sends application specific commands to invoke a trusted services. At the end of the communication, the session is shut down and TA either remains in an executable state where sessions can be set up with other CAs and TAs or returns to a locked state. In a locked state CAs and TAs cannot communicate with a locked TA because updates to codes and personalization information or general maintenance is performed on the TA by the Security Domain without the risk of any modification to this update by a Client Application or other Trusted Applications. In the next section covers a brief description of some of the pivotal TEE Internal Core and TEE Internal APIs as defined in the standard [83, 9].

### 3.3.1 TEE Internal Core API and TEE Internal API

The TEE Internal Core API provides a specific set of APIs providing functionality to the Trusted Application, enabling TAs to make use of Standard TEE capabilities [83]. The TEE Internal API on the other hand is a general series of APIs that provide common implementations for additional low-level functionality usually required by TAs. Some of the common APIs provided by the TEE Internal Core API as defined by GPD standards [9, 83] include:

1. Trusted Core Framework API: This API provides OS functionalities such as integration, scheduling, communication, memory management, and system information retrieval interfaces.

2. Trusted Storage API for Data and Keys: Trusted Key Storage and storage of general data is provided by this API.

3. Cryptographic Operations API: Cryptographic capabilities are provided by this API such as key generation, key exchange algorithms, Message Authenticate Codes (MACs), asymmetric encryption, asymmetric signatures etc.

4. TEE Arithmetical API: For cryptographic functions not found in Cryptographic Operations API, TEE Arithmetic API then provides arithmetical primitives to create these cryptographic functions.

5. Peripheral API: This API enables a Trusted Application to interact with peripherals via the Trusted OS.

### 3.3.2 TEE Internal API

We now discuss some of the APIs provided by TEE internal API as defined in [83].

### TEE Trusted User Interface API (TUI)

The TEE Trusted User Interface API gives the user assurance that what he sees on the screen has not been concealed, modified or accessed by another applications running on the REE, another unauthorised application in the TEE or a malware running on the device. TEE TUI also ensures that any information inputted by the user cannot be retrieved or altered by any application within the REE or even an unauthorised Trusted Application within the TEE. In addition to this, there is a security indicator, with which the user is assured that that screen displayed is actually a screen displayed by a Trusted Application. TEE TUI opens up a session during which TA has exclusive access to the User Interface, secure peripherals and prevents multiple sessions from being opened when this is happening

Furthermore, TEE TUI gives assurance to external parties that the information it receives from the user is exactly what the user saw, signed and it has not been interfered with along the way. Figure 3.3.2 shows a TUI architecture and some of the trusted components that exists in the TEE part of the architecture.

### Biometric API

This is an extension of the TEE Trusted User Interface API. Trusted Applications through the Biometric APIs can access biometric capabilities and functionalities present in Biometric Subsystem which is comprised of the Biometric Peripherals and Biometric Sensors. As a first step, using standard discovery techniques in the Peripheral API, the available capabilities in the platform is discovered during communication with the Biometric Sub-system of the TEE.

The Biometric sensors display live images, the relying TA cannot communicate with the Biometric sensors directly rather it interacts with the Biometric Peripherals and uses the service they provide once the biometric capabilities are known. The Biometric API is an important part of the TEE which is required for the eVoting Scheme proposed

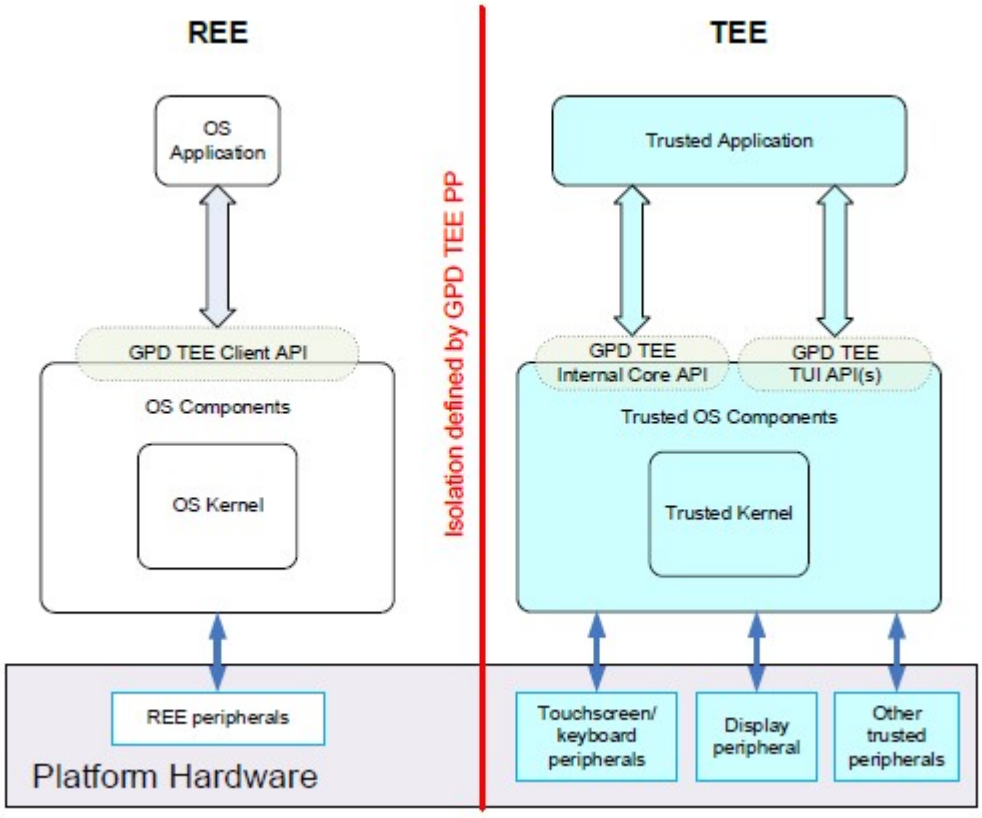


Figure 3.4: TEE Trusted User Interface [83]

in chapter 6

## 3.4 TEE Administration

A Global platform Security Domain has an off-device Service Provider that owns it and has the authority to directly or indirectly manage the Security Domains and Trusted Applications that fall under that SD. TAs and SDs have a direct association to a parent Security Domain that created it and an indirect connection to a SD along the path from the Root Security Domain in which its Parent Security Domain was created [5].

### 3.4.1 Security Domain

A Security Domain (SD) is a representative of the TEE Issuer, Original Equipment Manufacturer or any other Service provider, located on the device that is responsible for the control of administrative operations of the TEE.

The Security Domain can be pre-installed on the device during manufacturing of the device or can be dynamically loaded into the device by a service provider. Security Domains have privileges which gives them access to Internal TEE resources and hence provide provision of TEE properties [5].

Security Domains hold cryptographic keys which is used to commence the execution and authorization of administration operations in a secure manner. The level of privilege a Security Domain has would depend on the classification of the Security Domain. For example a Security domain can install/uninstall Trusted Applications and Security Domains if it has a TA Management privilege and SD Management privilege respectively [5]. Every device is allowed to have multiple security domains, even multiple root security domains. These security domains are configured to carry out authorization operations reflective of the roles and privileges of the authority it represents.

The Root Security Domain (rSD) [5], has the privilege of installing other security domains. Trusting a Security Domain requires a trust hierarchy, what that means is

that since we trust the rSD, then any other security domain created by rSD would also be trusted. A Security Domain that creates another security domain or trusted application is known as the Parent of that SD or TA because they have a direct connection to it. There also exist connections that are indirect for example a root SD might create a Security Domain, and this Security domain creates a trusted application. This particular trusted Application is said to have an indirect connection with the root SD. This implies that the Trusted Application is Trustworthy because its Parent Security Domain and Root Security Domain it has an indirect connection to are both trustworthy.

### 3.4.2 Administration of Security Domain

The GPD standard [5] states that Authorization Operations must be verified before any Administrative Operation is performed on TA or Security Domain. The recipient of the command to carry out a particular administrative operation might not be the SD to execute. This leads us to 2 concepts defined in the specification [5]:

1. The Authorizing Security Domain (SD-A) owns the credential required to verify the authorization. The SD-A also authenticates the Remote Authority that submitted the administration operation.

2. The Performing Security Domain (SD-P) is the SD that receives the operation command and performs the operation.In the Voting Protocol in chapter 6, an Election SD receives a command to create an Issuer SD, in that instance the Election SD is both the SD-A and SD-P

There are three options to authorize an administrative operation [5]:

1. Implicit authorization using a secure channel: for implicit authorization the Remote Authority sets a secure communication channel session with the recipient

SD which it intends to send an administrative command to. If this secure channel session is opened successfully then authorization command via this channel is verified. In implicit authorization, the Security Domain is both the recipient and performer of the operations i.e. SP-A=SP-D. In the voting scheme in Chapter 6, a secure communication channel is set up.

2. Explicit Authorization using Authorization Tokens: As discussed earlier, a Security Domain is a representative of a remote authority (Service Provider) on the device used to manage the TEE. A TEE has different security domains that may be representative of different remote authorities. To manage each SD that represents a remote TA, it may require having to set up individual secure channels to the Security Domains to implicitly authorize an administration command. Explicit authorization requires collaborations between these various remote authorities with different privileges to generate an Authorization Token sent to the SD-P that carries the administration command without having to set up individual secure communication channel sessions

3. Combined Authorization: A secure channel is set up and an authorization token is sent using this Secure Channel. So it's a combination of implicit and explicit authorisation. An Issuing Authority SD is created using this process in section 6.5.1

### 3.4.3 Authorization Token

As discussed earlier and Authorization Token is sent in an explicit authorization and an authorization token is a piece of information generated by a Remote Authority granting rights to its Security Domain located on the device to carry out certain administration operations. The Authorization token contains a Universal Unique Identifier (UUID) of the Security Domain that identifies the Remote Authority in charge of that Security Domain that emitted this authorisation.

It also contains a set of conditions called Constraints that need to be verified before executing the authorisation operations in the token. The Authorization Token also contains the Remote Authority's signature over the UUID and constraints. The token also contains the Key identifier that identifies the key to be used for the authorisation, algorithm identifier that identifies algorithm to be used to verify the authorisation.

Finally When SD receives the Token, using the Key Identifier to find the right public key of the Remote Authority stored in SD Secure Storage, it verifies the digital signature of the Remote Authority over the Token. This is shown in the Voting Scheme in Section 6.5.1, when Election SD creates an Issuing Authority SD.

### 3.4.4 Root of Trust

The TEE Root of Trust [8] is a computing engine code, data and keys, co-located on a platform that provides security services such as such as authentication, authorization, confidentiality, identification, integrity, measurement, reporting, update, or verification. A Root of Trust [8] verifies a software using a Chain of Trust where an already measured and verified software by the RoT, measures and verifies the next software and keeps recordable records of this verification. This process goes on until all the software are measured and verified forming a Chain of Trust. .

The TEE secure boot which is critical to its security uses a Chain of Trust. The first code that runs on the TEE is assumed to be Trusted, this code is hardware secure and confidentiality of the code provided using a key during manufacturing. The Boot Chain starts when this code is executed till it reaches the TEE Run Time Environment and validates this environment, the System-on-Chip prevents interference with this process. From then on any other stage in the boot chain sequence must be validated before it is executed [8].

## 3.5 Other Implementations of Trusted Execution Environment

The ARM Trustzone [23], Aegis [164] and Intel SGX [124] technology are implementations of Trusted Execution Environment Technology.

The ARM Trustzone, is a hardware implementation of a TEE. ARM Trustzone defines a Secure world where the security subsystems exist on the System-on-Chip which is Isolated from everything else in the Normal World by the Hardware Logic in the Trustzone-enabled AMBA3 AXI™ bus fabric [23]. Extensions implemented in the ARM processor core allows quick execution of codes from both the Secure World and Normal World by a single core processor.

Trustonic[3] is a security vendor that delivers secure devices and applications across multiple markets to different stake holders in the Ecosystem such as Mobile Network Operators and Application Developer.

Trustonic has developed the Kinibi TEE [166] compliant with the GlobalPlatform's TEE Initial Configuration v1.1 [6]. The Kinibi Trusted OS runs in the Trusted Execution Environment of an ARM TrustZone. Currently Trustonic's Kinibi TEE has been integrated in over one billion [166] mobile devices deployed worldwide.

## 3.6 Summary

In this chapter the architecture and security capabilities of the GlobalPlatform Trusted Exectuion Environment is presented. The TEE technology is pivotal in providing a trusted environment where security sensitive operations is executed in the mobile voting scheme proposed in chapter 6.

[3] https://www.trustonic.com/

# Part II

# Untrustworthy Environment

# I-Precinct Voting

# Chapter 4

# Analysis of Voting Schemes in The Real World

## 4.1 Introduction

Voting is at the heart of a country's democracy. Assurance in the integrity of the electoral process is pivotal for voters to have any trust in the system. Often, electronic voting schemes proposed in the literature, or even implemented in real world elections do not always consider all issues that may exist in the environment in which they might be deployed. In this chapter, we identify some real-world issues and threats to electronic voting schemes. We then use the threats we have identified to present an analysis of schemes recently used in Australia and Estonia and present recommendations to mitigate threats to such schemes when deployed in an untrustworthy environment

### 4.1.1 Overview of Assumptions in Electronic Voting Schemes

As democracies continue to grow, citizens of a lot of nations, more so in the developing countries, are beginning to clamour for the introduction of electronic voting because

they believe the traditional paper-based systems are often marred by wide scale electoral fraud [106, 67, 115, 95] One common issue with e-voting schemes is that the environment assumed during design may not fully consider the threats that exist in real world deployment. Thus, when these schemes are deployed some vulnerabilities may appear that were not considered in the initial threat model.

The voting environment and how voting schemes relate to other parts of the voting process goes a long way in determining which security requirements are necessary and which requirements may be satisfied by default. For example, a remote voting scheme and a supervised in-person voting scheme are two different voting environments and provide different levels of security by default. A supervised voting scheme can provide coercion resistance by being supervised but remote voting schemes do not give such guarantees by default. Consequently, remote voting schemes need to rely on technical security provided by cryptography.

With electronic voting, voter authentication is an open issue; it can be quite complicated authenticating the voter if done remotely. For example, a spouse may vote on behalf of her partner and there is no way the system can tell the difference. Some e-voting schemes have tried to address this threat by using smartcards [11, 78] as an instance of the voter. If the smartcard is authenticated as done in the Internet voting scheme used in Estonia [78] then the voter is assumed to have been authenticated correctly.

Many voting schemes might be suitable in one environment but unsuitable in another because of socio-economic factors (such as religion, poverty). These factors may determine how effective the voting schemes would be in such environments. For example, in a world where no one wants to cheat the system, we wouldn't have to worry about voters being coerced or ballot stuffing. It is well known that a system is only as secure as the weakest component. In a remote voting environment, while the network and servers are secure, there is no assurance that voter's computers are secure. In the

Estonian I-voting scheme, voter's computers were assumed to be secure. Subsequently, in a mock election [78], a group of researchers were able to attack voters' computers and change votes to their choice. So, to have assurance that the entire voting scheme is secure, the voter's computer needs to be secure as well.

Another common assumption made, is the trust placed on electoral officials. In precinct voting schemes, we trust electoral officials to correctly authenticate voters and prevent double voting [62]. However, this is not always the case in real world elections where the electoral officials may be part of the fraud [24]

After local elections in 2015 and parliamentary elections in 2017 were monitored, some recommendations were suggested to improve on the current methods of authenticating voters. The UK government agreed with most of the recommendations suggested and are now considering various methods for authenticating voters to prevent ineligible voting and voter impersonation. In some UK elections, pilot schemes were introduced using various forms of physical identification such as driver's license to authenticate voter. The essence of this pilot schemes was to gather information to find out what would best for the electoral process before adopting an official policy on voters and proxy voters authentication. We talk more about some of the recommendations made in section 4.1.2

We must note that the UK does not currently use electronic voting for elections and electoral fraud has not been reported on a wide scale that could influence elections, but it is interesting to see that these reports acknowledge the possibility of this being an issue if urgent changes are not made to the current system. If these recommendations are to be adopted in an untrustworthy environment section 5.2, then a cryptographic means of implementing some of these recommendations would be required. Moreover, to build a secure e-voting scheme, security must be considered at the outset and designed into the system. Security of all software and hardware should be analysed and proved secure if possible

### 4.1.2 Reports on Election in the UK

"My work in the Department for Communities and Local Government during the previous Parliament highlighted some shocking issues and revelations: our well-respected democracy is at threat from unscrupulous people intent on subverting the will of the electorate to put their own candidates into power, and in turn, manipulate local authority policy and funding to their own self-centred ends. That is something that we must do our utmost to guard against and to have measures in place to discourage and prevent". Erick Pickles [151]

The Anti-corruption champion, Erick Pickles monitored elections in the UK and made some recommendations [151] on improving elections in the United Kingdom, similar issues exists in other electoral systems around the world that rely on trust in human procedural processes to prevent voter impersonation and ballot stuffing. He went further to say that the idea that electoral officials can prevent voter impersonation by identifying the local populace through physical recognition is not sufficient [151] because our society is constantly growing with a lot of migration and emigration going on.

The Electoral commissions (EC)/ Association of Electoral Administrators (AEA), report [138] on UK Parliamentary elections in 2017 also echoed some of the issues raised in the Pickle's report, particularly relating to voter authentications and proxy voting.

The UK government in a response to EC's report and Pickle's report says it has adopted most of the recommendations [141]. Pilot schemes are been run in some polling stations introducing voter authentication, using various forms of Identity in a bid to gather enough information to see what best works before adopting an official policy for authentication for Voters and Proxy Voters [141]. Below are some of the recommendations suggested in the Pickle's report relating to Trust and voter authentication [151]:

1. Actions should be taken to allow voters go the polling booth individually and also prevent voters from taking ballot papers away from the polling units.

2. Government should consider using means of identification such as driver's license, international passport, and utility bills for voter authentication. Currently this is not done and that is unsatisfactory.

3. The registration phase needs to include automated means of confirming nationality of voters and their eligibility to vote because not everyone who resides in the UK has voting rights. If this is not done, one cannot prevent ineligible voting.

4. Proxy voters should also be authenticated using standard means of identification such as driver's license, international passports and utility bills.

5. Political campaigners and activists shouldn't be trusted to handle completed postal votes and postal votes envelopes.

6. Request for a waiver to provide signature for a postal voter, should require attestation by a third party and there should be restrictions on who can act as a third party.

7. The use of Camera phones should be prohibited at polling units because voters can record how they have voted and hence could be coerced to sell votes

## 4.2 Threat Analysis in Real World Voting Schemes

In other to build a secure voting scheme in a particular environment it is imperative to understand some of the factors that may affect the trust placed on different entities in the electronic voting architecture as well as threats that may exist in the environment the scheme would be deployed. With this in mind and looking at the recommendations in sections 4.1.2, we now consider issues that may affect the integrity of elections in real world implementations if not considered during the design phase of e-voting schemes in this section.

### 4.2.1 Socio-economic Issues

It has been documented that vote buying and vote selling is very prevalent in real world electronic voting. In Mexico, voters were so suspicious about the integrity of elections because of the electoral fraud committed by parties [67]. Such fraud relied on many techniques including ballot stuffing by both voters and electoral officials; stealing of ballot boxes between the polling units and collation centres; intimidation of voters, observers and party officials; and manipulating voter's registration lists [77, 24, 95] Vote buying, selling and coercion is common practice in elections.

In an analysis done in Taiwan [136] as little as 10 USD was paid to voters to sell their votes. This is not surprising because of the economic situation in many countries, and vote buyers usually target poor voters. In the USA five Democratic Party Operatives were convicted in a federal court in 2004 for offering poor people cigarettes, medicine, beer and 5 to 10 dollars for their votes [135].

In other cases, electoral officials are part of this electoral fraud. A report about the 2012 elections in Ghana recorded issues like double voting, under age voting, over voting and voting by ineligible individuals [24] this was possible because the pollsite officials were trusted to prevent this. These issues are difficult to address solely by human supervision because the trusted polling officials are sometimes part of the fraud, usually for financial gain. Voting schemes cannot prevent all forms of electoral fraud since there is always a financial incentive to cheat the system due to socio-economic challenges. However, design of voting schemes should take these threats into account and leverage on technical security wherever possible to ensure that any deliberate attempt to circumvent the technology is detected

### 4.2.2 Insider Threat

According to Schneier [161] "Insiders are especially pernicious attackers because they're trusted. They have access because they're supposed to have access. They have oppor-

tunity, and an understanding of the system, because they use it or they designed, built, or installed it. They're already inside the security system, making them much harder to defend against." The UK Cyber strategy also notes that "Computer systems, networks and applications all rely upon people for their development, delivery, operation and protection and the likely success of an attack is increased when a so-called 'insider' is involved" [140]

The insider threat is a well documented issue and one of the biggest threats to organizations. About 53 percent of attacks on organization have been deliberate actions or negligence by staff. 54 percent of IT staff feel it is difficult to detect insider threats while 33 percent of organization have no formal response plan [52].

Attackers have realized that it is difficult to attack secure networks, so they find easier routes, like targeting individuals that work in organizations. An example is the 2011 attack on RSA secureID[1] where phishing emails with an attachment that contained malware was sent to a group of unsuspecting employees who downloaded the files allowing the attackers to gain access to the network .

In e-voting literature, the insider threat and how it could mar an election is not often considered or it is assumed that insider can be trusted, hence human procedural means are used over technical means to carry out security sensitive operations. As an example, in the Prêt-à-Voter scheme, electoral officials are trusted to authenticate voters and prevent ballot stuffing [62] whilst in the i-voting scheme electoral officials are trusted to transfer of sensitive information from one entity to another [78].

In an analysis of the electoral process in Estonia [78], researchers recorded various lapses in procedures which introduced vulnerabilities that could be exploited. The financial benefits for malicious insiders is enough incentive for them to either aid an attack or look the other way when this happens.

With vulnerable electoral officials, it is important to ensure that the technical se-

[1] https://www.wired.com/2011/08/how-rsa-got-hacked/

curity employed in voting schemes should reduce threats posed by insiders. Hence, auditability of the process and verifiability of votes cast should be satisfied for a voting scheme to be credible.

In section 4.3 we do an analysis of the vVote: a verifiable voting scheme (Prêt-à-Voter) and I-voting scheme to shed more light on this issue.

### 4.2.3 Cyber Threat and Foreign Government Influence

Cyber threat and cyber warfare has become a serious issue that organizations and governments are dealing with. There have been various reported cases of state sponsored attacks like the alleged North Korean attack on Sony[2] or alleged United States attack on Iranian nuclear enrichment plant [114]. Increasingly we continue to see allegations of foreign government influence in the democratic processes of other nations.

In addition to the current controversy surrounding recent elections in the USA, it has been alleged that Russia carried out a state sponsored Distributed Denial of Service (DDoS) attack on Estonia in 2007[3] . In Hong Kong, the largest and most sophisticated ever DDoS attack hit an online democracy poll that canvassed opinions for future elections in the country[4] . Also, in Ukraine[5] a virus that was meant to delete votes during the presidential elections hit their Central Election Authority.

In Washington DC, an Internet voting system was designed to allow oversea absentee voters cast their votes, this was a pilot project and it was tested as a mock election in 2010. Some researchers [172] attacked this system and gained full access within 48 hours, changing every vote and revealing almost all secret ballots.

These Cyber-attacks have created a completely different threat environment that did not exist before, and now that nations are pushing for e-voting this should be considered when designing e-voting schemes.

[2] https://reut.rs/2VAiTye
[3] https://www.theguardian.com/world/2007/may/17/topstories3.russia
[4] http://bit.ly/2PBxvYP
[5] https://on.rt.com/elvysd

In the literature, schemes may not fully consider the threat of a cyber-attack from a foreign actor. In the I-voting scheme used in Estonia, lapses where shown in the electoral process and architecture that could create avenue for a cyber-attack [78]. The implicit trust placed on voters' computers in some Internet voting schemes clearly shows that cyber threat was not considered in their design.

### 4.2.4 Threat Model

Considering the threats/issues that exist in real world election discussed in section 4.2 and Pickle's recommendation (section 4.1.2), we now highlight some specific threats in figure 4.1.

Table 4.1: Threat Model Real World Elections

| Threat | Vulnerability | Impact | Scheme |
|---|---|---|---|
| Poll worker vote on behalf of abstained voters | Trust placed on poll workers to authenticate voters using traditional means.<br><br>Ballots are not authenticated, hence not digitally linked to voter. | Votes are cast for abstained voters without being detected by the system and could change the outcome of the election. | Prêt-à-Voter |
| An attacker can stuff the ballot without being detected. | Ballots are not digitally linked to voter | Ballot stuffing by poll workers and voters without being detected could change the election outcome. | Prêt-à-Voter |
| Poll workers can allow ineligible voting | Trust placed on officials over technical means to authenticate voter | Ineligible citizens can cast ballots undetected by the system, compromising election integrity | Prêt-à-Voter |
| An attacker can install a vote altering or data stealing malware in election servers and voter's computers | Unclean computers used to prepare voting client software sent to voters | Attacker alters votes to that of his choosing without being detected. Spyware monitors how voters voted, breaking ballot secrecy and could enforce voter coercion | I-Voting |
| Vote selling to coercers by voters | Trust placed on voter to tear candidate list that links ballot to voter | Voter can leave poll-site with candidate list and show a third party there by breaking privacy, receipt-freeness and coercion resistance | Prêt-à-Voter |

### 4.2.5 Attackers Capability

Below are some assumptions about the attacker based on reported incidences of electoral frauds reported in various elections discussed in sections 4.2 and threat model in figure 4.1.

- An attacker can either be an insider or an outsider.
- An attacker may be motivated by financial incentives to cheat the electoral process.
- A voter may be motivated by financial incentives to cheat the electoral process

**The attacker is assumed to have the following capabilities:**

1. An attacker can stuff the ballot box without being detected.
2. An attacker can vote on behalf of an abstained voter or allow ineligible voting.
3. An attacker has adequate resources to carry out a DDoS attack
4. An attacker can tell the link between a voter's id and the cast ballot.
5. An attacker can install vote altering, or data stealing malware in election servers and voters' computers.

## 4.3 Review of Two Electronic Voting Scheme

In this section, two electronic voting schemes are reviewed in the light of the threats and requirements discussed earlier.

### 4.3.1 vVote: A Voter Verifiable Voting Scheme (Prêt-à-Voter)

The Prêt-à-Voter voting scheme is an end-to-end verifiable scheme that provides privacy. It uses a candidate list which is printed on demand before voting. This candidate

list has the names of candidates arranged in a random order. The voter can audit this candidate list to confirm that it has the correct encryption of the random arrangement of the candidates. If this audit is done, the candidate list is decrypted and hence cannot be used to cast a vote to maintain the secrecy of the ballot [34]. A QR code of the candidate list is scanned into a tablet and this launches the vote capture application. The voter fills the ballot using the tablet. After completing the ballot, a preference receipt (PR) is printed. The voter can compare the candidate list with the preference receipt to confirm that they are both arranged in the same order, this gives the voter assurance that his vote has been cast as intended. The candidate list is expected to be destroyed after confirmation, to maintain ballot secrecy while the preference receipt is kept by the voter. At the end of the elections, a voter uses the preference receipt to confirm that his vote has been published on a Web bulletin board at the final tally. This gives voters assurance that their votes have been recorded as cast [34]. Further details could be found in the draft report [159]

### 4.3.2 Estonia Internet Voting System (I-voting)

Over 30 percent of votes cast in elections done in Estonia today are done electronically this makes Estonia one of the front runners in the use of electronic voting for elections and the first country to use Internet voting nationally [78]. Estonia has a national ID card which has cryptographic keys issued by the government which are used to authenticate voters during election. The scheme attempts to replicate the double envelope process used in postal voting. A digital signature is generated with the voter's signing key and this is used to provide the voter's identity (outer envelope). The system's public encryption key is used to encrypt the ballot to provide secrecy (the inner envelope). The signature is stripped from the ballot leaving a set of anonymous encrypted votes once the eligibility of all voters has been established. These anonymous votes are then transferred to a physically separate vote counting server connected to a hardware security module for decryption.

The Estonian voting system uses a vote forwarding server which is the only publicly accessible server (see figure 4.1). This server communicates with the client software and forwards vote to a vote storage server. Votes are copied using DVDs to the Vote Counting Server by electoral officials. The Vote Counting Server is not connected to any server. The Estonian Internet voting system is not end-to-end verifiable and much of security it provides relies on human procedures rather than technical means thereby placing lot of trust on electoral officials. Further details about how this voting scheme really works can be found in [93]

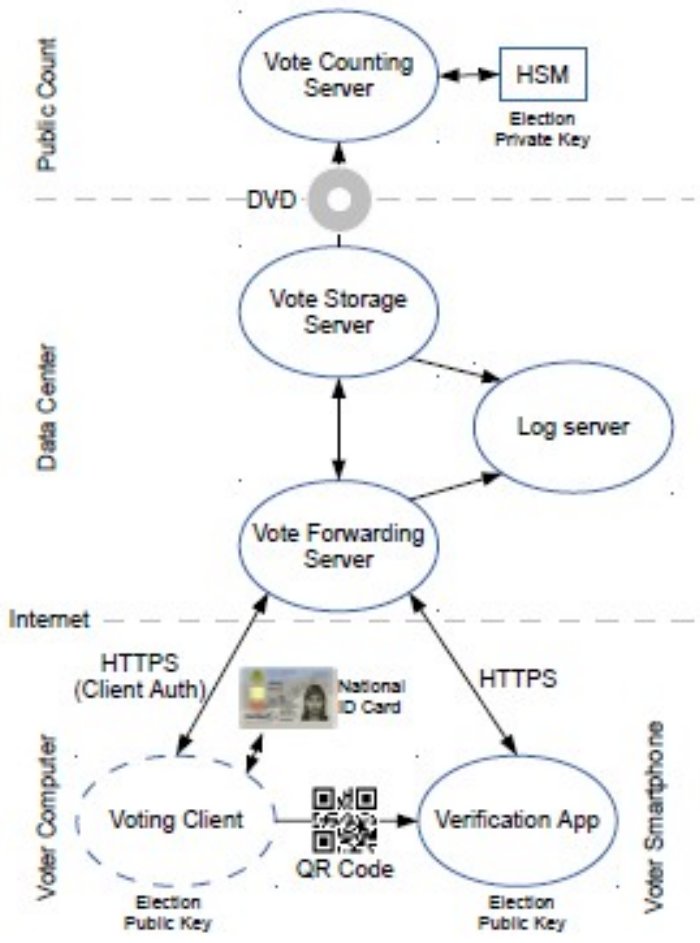


Figure 4.1: iVoting Protocol [93]

### 4.3.3 Security Analysis of vVote: A Verifiable Voting Scheme (Prêt-à-Voter)

Prêt-à-Voter, used in Australia, relies on traditional means to validate eligibility. If this scheme is to be adopted in other environments, this may not work since part of the reason why nations clamour for electronic voting is the inadequacies of traditional means of authenticating voters. As stated in our threat model, relying on poll workers to authenticate voters could leave the system vulnerable to ineligible voting. This risk could be mitigated using Photo ID but cannot be eliminated by this approach alone especially if the poll workers are untrustworthy.

This scheme expects voters to destroy the human readable candidate's list after casting their votes, this puts a huge level of trust on voters to do this. Privacy is an important requirement of e-voting as well as receipt freeness which helps to mitigate vote buying, vote selling and voter coercion. If voters fail to destroy this human readable part, which isn't unthinkable considering the socio-economic challenges (Section 3.1), then this scheme would not provide privacy. Because with this candidate list you can make a link between the candidates and vote cast published on the bulletin board as highlighted in our threat model is section 4.2.4.

Furthermore, with the candidate list, voters have proof to show a vote buyer or a coercer. Thus, this scheme would fail to provide coercion resistance and gives voters the opportunity to sell their votes to vote buyers. Ballot stuffing, double voting and voting in place of abstained voters could go a long way in determining who wins an election and has been reported in several elections, an example is the 2012 national elections held in Ghana [24]

The Prêt-à-Voter scheme is as vulnerable to corrupt official as traditional schemes and this needs to be mitigated using technical means. A corrupt official could vote for an abstaining voter and this wouldn't be detected by the system because of unlinkability between voter and the ballot cast; and lack of technical means for authentication.

Schemes like the I-voting in Estonia solve this problem by using a smartcard which is an instance of the voter. This link prevents corrupt officials from voting for abstained voters without physically having their smartcards. Prêt-à-Voter system is meant to be end-to-end verifiable but the attacks mentioned cannot be detected by the system and represent a big risk to take in certain environments. Hence, considering insider threats and socio-economic issues like poverty, Prêt-à-Voter may not offer any better security than traditional schemes.

However, Prêt-à-Voter would improve efficiency and minimize human errors in the vote counting process and if there is assurance that the electoral officials and voters are trustworthy, then it may well satisfy its security claims. However, this is easier said than done in the real world.

In conclusion, Prêt-à-Voter is vulnerable to breach of privacy, vote buying and selling since achieving receipt-freeness relies on voters being trustworthy, ballot stuffing, ineligible voting is possible if trust assumptions are broken.

### 4.3.4 Analysis of Estonian Internet Voting Scheme

A group of researchers observed the Estonian elections and produced a report which showed lapses in the electoral process that could undermine the integrity of the election. One of the issues raised was the use of procedural means over more technical means to provide security. A high degree of trust, as seen in Prêt-à-Voter system, was placed on electoral officials, making security critical aspects of the system rely on, sometimes, a single individual. Trust was also placed on the integrity of voters' computers as well as the various servers used. Some of these lapses are considered to support our argument but a full report on this election can be found in [78, 93] .

Contrary to security best practices electoral officials logged on to servers using root access. This is a major lapse because the system cannot tell which official accessed it. This creates an opportunity for a malicious insider to carry out attacks such as

installing malware that could alter votes between decryption and tabulation, or stealing information that could compromise vote privacy as highlighted in our threat model in section 4.2.4

It was also observed that the vote storage server reported an error suggesting that the drive configuration had changed when it was booting during the tabulation phase. Instead of the officials investigating this error, it was simply bypassed in this critical phase of the election where encrypted votes are exported. In other instances, servers were simply rebooted to clear error messages rather than troubleshooting. If these errors were caused by malware, the officials would not have noticed. And since the system is not end-to-end verifiable, voters and auditors cannot tell if votes are counted as cast in the final tally.

It was also documented that officials downloaded client software using an unsecure http connection. This makes the system vulnerable to a network man-in-the-middle attack which could compromise the election. Unclean laptops that had links to gambling sites and bit torrents installed were used to prepare client software distributed to the public, this could introduce malware into voter's computers (section 3.4) on a large scale. Most attacks on organizations are carried out because unsuspecting insiders are targets of cyber attackers. So even if the electoral officials are not genuinely part of the electoral fraud, their actions as have just been highlighted leaves the electoral process vulnerable to attacks.

In the Estonian system [78], a voter can verify with an application that their vote was cast as intended. However, with the increasing interaction between smart phones and computers it is not difficult to imagine that both devices can be corrupted making it difficult for the voter to notice that their votes have been altered.

In the tabulation phase, it was also reported that a technical glitch occurred and an official's personal flash drive, that contained other personal files, was used to copy unencrypted votes to a laptop connected to the internet where the official result was

signed. If this USB contained malware, this would mean the votes could have been altered without detection. Furthermore, the flash drive could have introduced malware to the counting server, this malware could be a spyware which could have the ability to monitor the decryption process and hence know the relationship between a voter and a ballot- breaching voter's privacy. These possibilities were identified in our threat model in section 4.2.4

From the published portions of the I-voting server software the researchers found out that the log server, which logs information from the vote forwarding and vote storage servers, saved any unexpected data to disk. If this storage gets exhausted voting would stop allowing a denial of service attack Such an attack is well within the means of the state sponsored attacker or even a modest attacker with adequate resources (section 3.4) depending on the size of the disk. Moreover, storing of unexpected data means the system is vulnerable to other attacks.

In conclusion, we can see that this scheme is vulnerable to many attacks such as DDoS, breach of privacy, and vote alteration. Some of these attacks are possible because the scheme is not universally verifiable (section 2) and trust was placed on human procedures and processes rather than technical security.

## 4.4 Further Analysis

It is clear that while existing e-voting schemes may be secure in benign environments, their adoption for use in untrustworthy environments presents a number of risks. Section 4.2.4 identified the threats to e-voting schemes and how these could impact security. In this section, a further analysis on the schemes considered in section 4.3 was done, highlighting the motivation for the attacker; Vulnerabilities that could be exploited and some ways to mitigate the threats.

- **Authentication:** In the Prêt-à-Voter scheme, its been shown that trust placed on electoral officials may allow ineligible voters vote, over voting and voting on

behalf of abstained voters. These vulnerabilities could be prevented by the introduction of a tamper-resistant token, such as a smartcard which is difficult to clone. With smartcards voters can be authenticated correctly preventing ineligible voters from voting. Such a device could also sign ballots ensuring linkability between the voter's id and cast ballot. Linkability would ensure officials cannot vote for absent voters without the smartcard in their possession, preventing ballot stuffing.

However, the use of smartcards comes at an extra cost and the added advantage of introducing a smartcard may not be justified. The Estonian National ID card which has cryptographic keys for both authentication and digital signature adequately addresses the issue of ballot stuffing and voter's authentication. There also exist other electronic voting schemes that use smartcards for authentication of voters [11], prevention of double voting and impersonation of abstained voters.

- **Incentives:** In the Prêt-à-Voter scheme, the trust placed on voters creates an opportunity for vote buying, vote selling and coercion because of the possibility of voters carrying the human readable candidate list out of the polling booths. This means receipt-freeness, which underpins privacy and coercion-resistance, relies on voters who may not always be trustworthy. We have shown that considering the socio-economic issues in societies, there is an incentive for voters and electoral officials to partake in corrupt electoral practices. In many environments, a voting system cannot rely solely on trustworthy voters or electoral officials but should rather rely on technical measures for security sensitive processes.

  In the I-voting scheme, the procedural lapses highlighted in section 4.4 creates an avenue for a malicious insider to infect voter's computers on a large scale. Personal devices should not be used to prepare client software sent to voters. Special purpose laptops or PCs dedicated to this task should be used. Moreover, the integrity of any software should be checked at intervals to ensure that the

client software has not been tampered with.

- **Verifiability:** The I-voting scheme places trust on the voter's client machine. Malware can be introduced to the system by taking advantage of procedural lapses highlighted in section 4.4 above. This kind of attack would go unnoticed by voters and auditors because the scheme is not end-to-end verifiable (verifiability section 2). Trust was placed on the voter's client machine, human process and procedures to prevent this. Our argument is that such trust is misplaced, thus verifiability is difficult to guarantee.

  Further attacks on the I-voting system were carried out on both the client side and server side affecting ballot secrecy and voter's privacy [78]. The Prêt-à-Voter scheme, however, is end-to-end verifiable from vote casting to the final count and tallying of results. Any vote alteration would be detected both by voters and third parties because this scheme relies on sound security practices to provide verifiability rather than relying on human processes and procedures.

- **Cyber Attacks:** We have shown that Cyber attacks by well-resourced attackers or state sponsored attackers are a threat to electronic voting which did not exist in traditional paper based schemes. The DoS attack that could stop voting in the I-voting scheme by exploiting data logging (see section 4.4) can be prevented by ensuring proper validation of input data. Furthermore, the use of unclean laptops to prepare client software sent out to voters creates an avenue for wide scale malware infection of voters' computers. This kind of vulnerability could leave the entire e-voting process vulnerable to a full scale cyber attack. In the Prêt-à-Voter scheme the voting is done in a more controlled environment since elections are done in a precinct and all the equipment used are under the control of the electoral authorities.

- **Technical Security vs Human Procedures and Processes:** Many technical

solutions to ID verification rely on PINs or passwords. These may be stolen, or shared. Alternatively, a physical token could be issued that generates temporary passwords or PINs. This gives an extra level of security because even if PINs have been compromised an attacker would still require a physical token to make use of that information. We argue that the cost of issuing such tokens may be justified depending on the environment where the e-voting scheme is deployed.

We have shown from both schemes analysed, that the part of the scheme where technical means are used to provide security and the kind of trust assumptions made could determine which security requirements would be satisfied. In the Prêt-à-Voter scheme, from vote casting to the final vote tallying is end-to-end verifiable but the Estonian Internet Voting scheme is not.

This gap in technical verifiability between authentication and vote casting in Prêt-à-Voter, and in the I-voting scheme between vote casting and vote tallying, introduces weakness which could be exploited by an attacker as discussed in sections 4.3.3 and 4.3.4 respectively

In the Prêt-à-Voter scheme authentication was undertaken by traditional means since it is a supervised scheme, this is a sharp contrast from the I-voting scheme which used a smartcard to authenticate voters. The lack of technical authentication in Prêt-à-Voter creates the opportunity for electoral officials to cheat the system as discussed in section 4.3.3 without being detected.

On the other hand, the trust placed on voters to destroy the human readable part of the Prêt-à-Voter ballot form used to verify votes are cast as intended, creates another vulnerability that could be exploited by voters to break ballot secrecy and sell votes. In the I-voting scheme, a verification app on a smart phone is used to check that votes have been cast as intended. However, in this scheme, a coercer can watch a voter while voting. This could be mitigated by a re-voting option to

override any coerced vote. In cases where the voter is corrupt, no technology can prevent the voter from selling votes or allowing someone vote in his stead because this scheme is not supervised.

In conclusion, both e-voting schemes reviewed have advantages and disadvantages in terms of meeting the requirements for a secure e-voting scheme. However, neither scheme is able to meet all requirements. In particular, we have identified two issues that need to be addressed if e-voting schemes are to be used in untrustworthy environments. Firstly, methods to mitigate threats posed by insiders are required; and secondly robust methods to authenticate the voter needs to be addressed. These issues need to be considered in any practical implementation of voting schemes if they are to be widely deployed.

## 4.5 Summary

Electronic voting systems are beginning to move from the lab to real world election. Such systems have many potential benefits, however, at this stage there are some impediments that may leave the systems vulnerable in an untrustworthy environment.

In section 2, a set of threats were identified and a threat model was built. We used these threats to analyse two e-voting schemes that have recently been used in real world elections to consider how they would fare in an untrustworthy environment. The analysis of these schemes highlighted the difficulty in preventing coercion resistance, vote buying and vote selling in the remote voting scheme, since a voter can be easily be impersonated or monitored whilst voting. In supervised voting schemes, analysis of Prêt-à-Voter establishes the fact that poll workers cannot be relied on to authenticate voters using traditional means in an untrustworthy environment.

We argue that in order for a voting scheme to be deployed in untrustworthy supervised voting environments, voter authentication and ballot authentication should not be external to end-to-end verifiable voting schemes. If eligibility verifiability is external

to the voting protocol, it creates a gap in technical security that the system cannot verify if the human procedural trust assumptions do not hold. With this in mind, fewer trust assumptions should be made about the honesty of voters, electoral officials and observers.

We appreciate that it is highly unlikely that voting schemes would completely eliminate human procedures and processes since security cannot be achieved by technology alone. However, we argue that where security could be provided by technological means, then this should be leveraged wherever possible in the electoral process. In chapter 5, an electronic voting scheme is proposed that addresses the issues raised in this section.

# Chapter 5

# Mutual Authentication Voting Scheme in an Untrustworthy Environment

*Trust assumptions in electronic voting schemes may not hold when deployed in certain environments. Authentication of the cardholder and ballot can prevent voter impersonation and ballot stuffing by a particular cardholder. In this chapter we propose a generic mutual authentications scheme, using smartcards and biometrics to authenticate the Voter. We also do a security analysis to show how this scheme satisfies our protocol objectives. We then go further to incorporate a mix-net into our generic mutual authentication scheme and show the scheme satisfies security requirements of an electronic voting schemes*

## 5.1 Introduction

Eligibility of the voter is a pivotal security requirement of any e-voting scheme. If the scheme cannot correctly identify voters, then it cannot prevent voter impersonation or

double voting.

In many precinct electronic voting schemes, such as prêt-a-voter [62] traditional means such as those used in paper based elections are used to authenticate voters. This places a high level of trust on both voters and polling station officials to act honestly. While this may be reasonable in a benign environment, in a high coercion, untrustworthy environment, stronger methods of authentication needs to be introduced, this issue is discussed in details in chapter 4.

In elections conducted in countries like Nigeria, Ghana, Algeria etc, biometric authentication is used to authenticate voters in a bid to reduce ineligible voting.These countries have poor documentation of citizens which is part of the reason why they have opted for biometric voter registration to ensure only eligible voters appear on the voters register. However, the actual vote casting is done using traditional paper based secret ballots (Australian ballots). This creates a disconnect between voter authentication and vote casting, which gives an opportunity for electoral fraud to be perpetuated.

In our generic mutual authentication scheme, we propose the use of a multi application smartcard or token with biometric verification capability to authenticate voters. The smartcard should have match-on-card biometrics capability with liveness checking (for example fingerprint biometrics [13]) using a template stored on the smartcard to prove that the voter is present during the transaction. We chose this approach because it is a technology already being used in many countries to authenticate citizens before access is granted to government applications and resources.

We also include the optional use of PINs as a secondary means of voter verification. PINs would be verified by the issuer of the smartcard rather than the card itself. This online PIN verification and biometric authentication provides a high level of assurance that the smartcard is presented by the voter it was issued to.

We incorporate our mutual authentication protocol into a re-encryption mix-net scheme for electronic voting and show how the scheme provides anonymity of the voter whilst still providing voter and ballot authentication. Finally, a security analysis to

show the protocol is secure and achieves its protocol objectives is conducted.

### 5.1.1 Biometric Authentication in Electronic Voting

In many countries across the world, most especially in developing countries, poor documentation of citizens is prevalent. According to the Centre for Global Development (CEDG) about 2.4 billion people[1] do not have widely recognised means of legal identification. The inability to properly identify citizens of a nation leads to electoral registers filled with errors which can then be manipulated to carry out wide-scale electoral fraud.For this reason, there has been a growth in the use of Biometric registration of voters and biometric voting systems for binding elections. The goal is to eliminate multiple enrolments on the eligible voter's list, prevent double voting and ultimately guarantee a free and transparent election. Furthermore, the use of Biometric authentication reduces the chances of voter impersonation, carousel voting and misuse of deceased voter's record because biometric technology is a reliable way to authenticate voters based on their unique physical characteristics.

In 2008 [173], Bangladesh registered 80 million voters using Biometric Technology just before the ninth parliamentary election, this election was observed and concluded to be the best election held in the country [99] . In 2015, 6 Million[2] voters were registered in Guinea using Biometric technology.In Nigeria, as at 2015, over 70 million voters[173] were registered for elections using Biometric Identification and voters are authenticated using biometric technology. In 2017, Somali Land conducted an election using Iris based biometric technology[3] to authenticate voters. Currently about 50 countries around the world have adopted biometric technology for voter identification at the polling stations according to election report[4] from the Institute of Democracy

[1] https://www.cgdev.org/blog/finding-missing-millions-identity-and-sdgs

[2] https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/identity/enrolment/biometric-voter-registration

[3] https://www.africanews.com/2017/11/14/somaliland-is-first-in-the-world-to-use-iris-biometric-voting-system-hi-tech//

[4] https://www.idea.int/data-tools/question-view/739

and Electoral Assistance (IDEA).

In our generic authentication protocol, we intend to build on the existing biometric technology already in use in these challenging environments.

### 5.1.2 National Electronic Identification Card Programs

Currently, over 70 countries have implemented National Electronic Identification Card (eID) programs. According to a report[5] by Acuity, there will be 3.6 billion National eID cards in circulation by the end of 2021 and in 82 percent of the eID programs implemented by countries, either Chip or Plastic Cards with Biometric was issued. These eIDs issued to citizens, allows the government authenticate citizens before access to government applications and electronic resources is granted. Electronic IDs have been implemented across countries in Europe, Asia and Africa[6]. In Estonia, citizens use their eID[7] as a legal travel ID for travelling within the EU, as a national health insurance card, for proof of identification when logging into bank accounts and as a voter's card for I-voting amongst other uses.

In Nigeria, as at 2019, 38 million National eIDs were issued to citizens[8]. As at June 2021, 60 Million citizens were enrolled and eligible for National eIDs according to the National Identity Management Commission (NIMC)[9]. The eIDs issued has 13 applets with 5 already activated for Electronic Identification (eID), EMV Payment, Biometric Match-on-Card Verification, Electronic Public Key Infrastructure (ePKI) and Travel – International Civil Aviation Organization (ICAO). The other applets are reserved for e-Voting application, e-Health, e-Transport amongst others.

Our focus for the proposed generic voting scheme is to build a voting protocol using the existing eID smartcards issued to citizens leveraging on the cryptographic and biometric verification capabilities of the card. We discuss more about the capabilities

[5] https://www.acuitymi.com/digital-identity

[6] https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/identity/2016-national-id-card-trends

[7] https://e-estonia.com/solutions/e-identity/id-card/

[8] https://www.thecable.ng/nimc-38m-nigerians-have-been-issued-national-id-number

[9] https://nimc.gov.ng/nimc-reaches-more-than-sixty-million-60-unique-nin-records/

of the smartcard in sections 5.3.1.

## 5.2 Untrustworthy Environment 1

As discussed in Section 5.1.1, in many developing countries, voter registration remains an extremely difficult issue which has been the most contested part of the electoral process [173]. In these countries, population census is not trustworthy [90][123], legal means of identification are unreliable thus voter registration is extremely complicated [173]. Hence, existing voter's register are of very poor quality creating an avenue for electoral fraud such as impersonation of voters, carousel voting, ballot stuffing and disenfranchisement of legitimate voters.This implies that trust assumptions that pollsite officials can properly authenticate voters without the use of technical authentication mechanisms may not hold in a high coercion environment

Another issue with some elections, is the Implicit trust placed on voting terminals used by voters to cast ballots. In certain environments it is reasonable to trust the voting machines but in challenging environments complete trust cannot be placed on these voting machines. In some schemes [11] these terminals are not authenticated by the voter. Which implies that in practical deployment, these trust assumptions may not hold, thus compromising the integrity of the election. In sections 4.2.1 we discussed some of the socio-economic issues that can motivate an attacker to participate in electoral fraud and why some otherwise secure systems may present vulnerabilities when deployed in these environments. Furthermore, it has been documented in several reports [135, 136, 115, 95] that pollworkers and voters may deliberately be part of electoral fraud to cheat the system for financial gain

With all the issues discussed concerning possibilities of electoral fraud, we define an untrustworthy environment in an e-voting context to be an environment where complete trust cannot be placed on poll-site officials, voters or voting terminals because they may have been compromised, either intentionally or unintentionally, due to

socio-economic issues and lapses in technical and operational security. With this in mind, extra technical and operational security measures is needed to reduce the risk of electoral fraud, or at the very least detect anomalies.

### 5.2.1 Threats in an Untrustworthy Environment

In light of the issues that exist in an untrustworthy environment, we suggest a list of threats as follows:

1. An attacker can impersonate a legitimate voter and vote on his behalf.
2. Malicious insiders, e.g. poll-site officials, may try to vote on behalf of abstained voters.
3. Malware may be introduced into voting terminals that could alter votes.
4. An attacker may control the channel between the voting terminal and smartcard or token.
5. Voters may try to participate in vote selling and ballot stuffing.
6. Voters may try to give or sell smartcards or tokens issued to them to third parties, to allow impersonation on election day.

### 5.2.2 Security Requirements

Given the threats highlighted in section 5.2.1 we now consider some security requirements for an e-voting scheme in an untrustworthy environment.

**Eligibility Verifiability** Only eligible voters are allowed to vote and cannot double vote.

**Receipt Freeness** The voter does not get any information that can be used to show how he voted to a coercer.

**Privacy** Votes should not be linked to voters, meaning no one should be able to tell how a voter voted.

**Universal Verifiability** This means a voter can verify if his vote has been cast-as-intended and recorded-as-cast. Any observers should be able to verify votes have been counted-as-recorded.

### 5.2.3 Authentication in Other Electronic Voting Schemes

Several e-voting schemes have been proposed that meet some of the above requirements.

In the FOO scheme [79], and in an implementation of the SENSUS scheme [59], it has been shown that election authorities can vote on behalf of abstained voters because the ballot messages are not authenticated by the voters [26].

The vVote scheme [62] trusts officials to correctly authenticate voters and prevent ballot stuffing. This creates a vulnerability corrupt officials could exploit to cast votes for abstained voters. Although vVote is end-to-end verifiable, the scheme cannot detect this because nothing ties a cast ballot to a particular voter's identity. In the election conducted using this scheme in Australia [159], the verification feature of the protocol was not used. This implies that voters were unable to verify the correctness of the ballots generated, we discuss this in detail in Sections 4.3.3. A dual ballot scheme [30], has been proposed and used in both a student election and political party elections in Israel but, in an untrustworthy environment, it suffers from similar eligibility verifiability weaknesses.

In Civitas [49], the voter has a registration key which is used to authenticate to the registrar but the protocol does not specify how exactly this happens and what messages are exchanged during this process. Also, the scheme does not specify how keys are managed by the voter [134]. The scheme was improved using a smartcard to manage the credentials on behalf of the voter but still does not specify what messages are exchanged during voter authentication.

The Secure National Electronic Voting scheme [11] issues smartcards [10] to voters and uses fingerprint biometrics for voter verification. The voting terminal encrypts the ballot filled by the voter and sends this to the voter's smartcard to sign. However, this scheme places trust on election workers to transfer sensitive information between various servers not connected a network. Moreover, if the voting terminal is compromised and alters the completed ballot, the smartcard would sign this modified ballot without the voter's knowledge. Also, the voting terminal authenticates the smartcard using the last ballot [11] cast and static data signed by the card issuer stored on the card. Static data for authentication creates weaknesses in the protocol that may be exploited to carry out other attacks, an issue that has been widely covered in EMV transactions [122, 130].

In a remote unsupervised environment, technical security cannot prevent a fraudulent voter from selling votes or a coercer monitoring voters cast ballots. Hence a more realistic approach is to conduct e-voting in a supervised environment, leveraging on technical security so that fewer trust assumptions are made. This is part of our motivation for proposing an electronic voting scheme that builds on the security capabilities of National eID program as discussed in Sections 5.1.1 and 5.1.2 that already exists in these untrustworthy environment, to provide an extra level of security to the electoral process.

### 5.2.4 Contribution

1. We propose a generic mutual entity authentication protocol that guarantees voter eligibility and liveness using biometric technology to minimize trust placed on polling station officials and equipment.

2. We show how our protocol can be integrated with a re-encryption mixnet to achieve voter anonymity, whilst still providing ballot authentication and increased security in an untrustworthy environment.

[10] The smartcard contains a private key for digital signature that never leaves the card
[11] Including the last ballot cast and ballot ID prevents an adversary from replaying ballots

3. We provide a formal verification of our protocol using Scyther and show that it satisfies eligibility verifiability which is a security requirements of an e-voting scheme

## 5.3 Generic Mutual Authentication Scheme

In this section we describe the various entities of the mutual authentication scheme. But first we describe the Multi Application Smartcard which will be issued to voters as a Voter's card

### 5.3.1 Multi Application Smartcard

A multi application smartcard as seen in figure 5.2, is a chip based smartcard with a microprocessor that can host several applets on the card, allowing card holders use a single card to carry out different transactions. This is part of the reason why many countries have chosen to issue citizens Multi App Smartcards for Electronic ID (eID). The different applets on the card run in dedicated and isolated memory areas, that cannot be accessed by other applets on the card. Special firewalls are used to enforce the applet separation so there is no interference of operations between different applications. As stated in Section 5.1.2, the eID used in Nigeria and Estonia can be used for payments, biometric authentication and electronic voting.

These eIDs meet high advanced security requirements and supports cryptographic algorithms such as RSA up to 4096 bits, elliptic Curves up to 521 bits, hash functions and symmetric encryption algorithms as seen in figure 5.1. These cards can also generate random numbers, encrypt messages, decrypt messages, generate digital signatures and verify digital signatures. These Multi app cards are Common Criteria EAL 5+ and EAL 6+ certified. In our proposed scheme, we make use of the existing cryptographic capabilities offered by the Multi App smartcard issued to citizens of countries in our defined **Untrustworthy Environment 1**

| Product | Product Description | Leading Technologies | Applications | Interfaces | NVM / [kByte] | RAM [kByte] | User ROM [kByte] | CPU | Symmetric Cryptography | Asymmetric Cryptography | Delivery Forms | Certifications |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SLE 78CFX3000P | security cryptocontroller | Integrity Guard SOLID FLASH™ | government identification | ISO 7816 | 300 | 8 | | Dual 16-bit | AES up to 256-bit DES, 3DES | ECC up to 521-bit RSA up to 4096-bit | contact-based module sawn wafer | CC EAL6+ high EMVCo |
| SLE 78CFX4000P | security cryptocontroller | Integrity Guard SOLID FLASH™ | government identification | ISO 7816 | 404 | 8 | | Dual 16-bit | AES up to 256-bit DES, 3DES | ECC up to 521-bit RSA up to 4096-bit | contact-based module sawn wafer | CC EAL6+ high EMVCo |
| SLE 78CFX5000PH | security cryptocontroller | Integrity Guard SOLID FLASH™ Mega Memory | government identification | ISO 7816 | 500 | 12 | | Dual 16-bit | AES up to 256-bit DES, 3DES | ECC up to 521-bit RSA up to 4096-bit | contact-based module sawn wafer | CC EAL6+ high EMVCo |
| SLE 78CFX6280PH | security cryptocontroller | Integrity Guard SOLID FLASH™ Mega Memory | government identification | ISO 7816 | 628 | 12 | | Dual 16-bit | AES up to 256-bit DES, 3DES | ECC up to 521-bit RSA up to 4096-bit | contact-based module sawn wafer | CC EAL6+ high EMVCo |

Figure 5.1: Smartcard Specifications

### 5.3.2 Entities, Data and Assumptions

We begin by describing the different entities involved in our protocol; the relationship between these entities; and the data stored on each entity. This is illustrated at a high level in fig 5.3

### 5.3.3 Key Entities

**Voting Authority (VA)** VA is a Certification Authority. It certifies the public keys of $IA$ and $VT$.

**Issuing Authority (IA)** Issues the Voter's Card (VC) and shares a symmetric key with $VC$. The issuer also stores the electoral roll and can check which voters have voted.

**Voters Card (VC)** This is a tamper-resistant multi app smartcard or token that

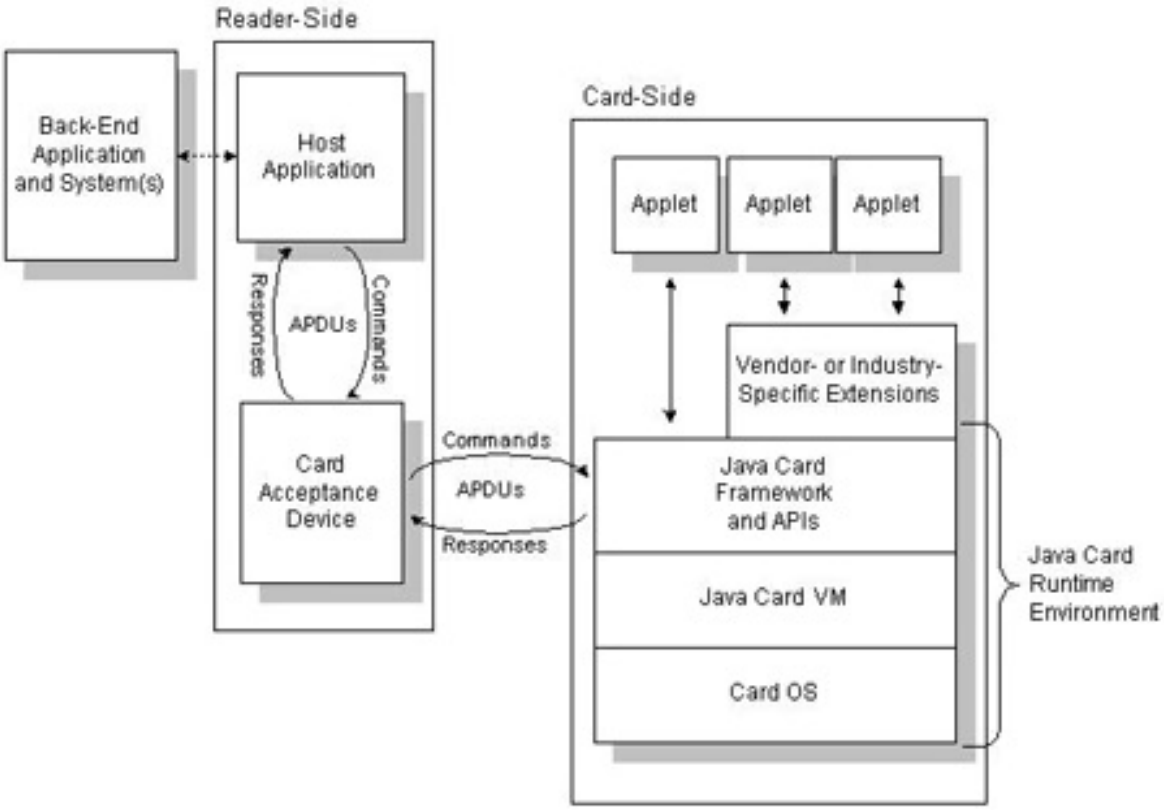


Figure 5.2: Multi Application Smartcard

carries out cryptographic functions on behalf of the voter.

**Voting Terminal (VTerm)** This is the complete voting terminal. Voting is done using this machine and it has a Graphical User Interface (GUI). VTerm is also equipped with a PIN pad, card reader and biometric capturing machine.

**Trusted Zone within VTerm (VT)** This is a Global Platform Trusted Execution Environment within the Voting Terminal. $VT$ is tamper resistant and secret information cannot be extracted from it. The $VT$ can generate random numbers, session key, encrypt messages, produce digital signatures and all these operations are carried out within $VT$.

In the case of $VA$ and $IA$, these are logical entities, included for scalability. In

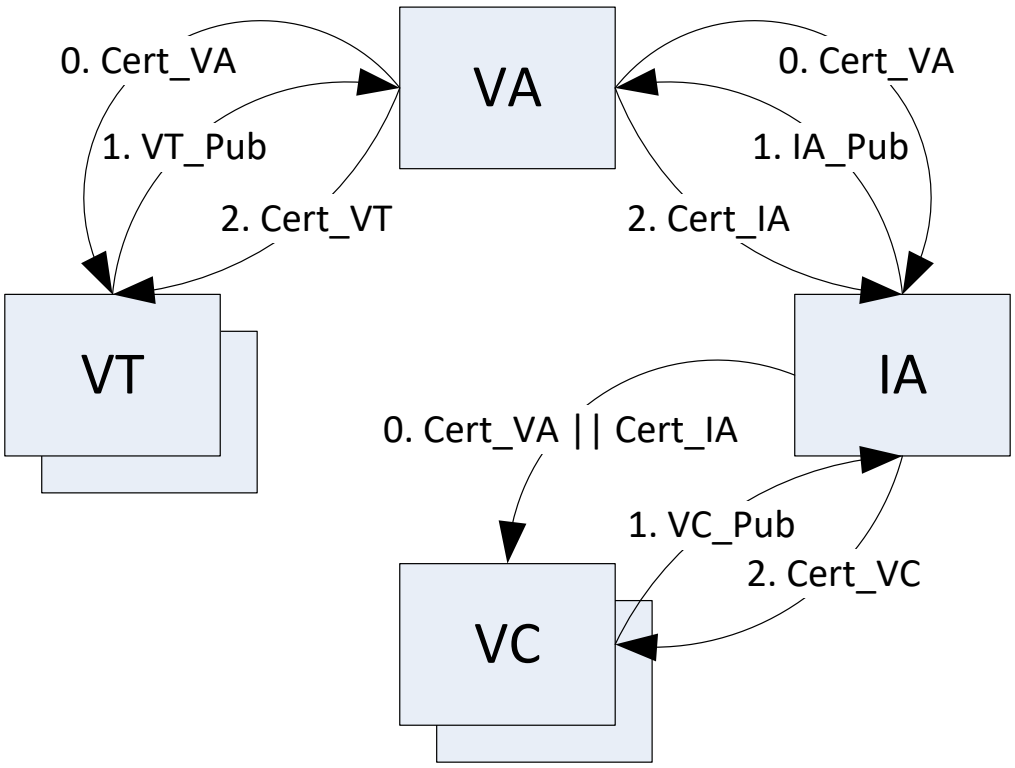


Figure 5.3: Key Entities

practice, in smaller elections, it may be possible to combine these roles.

An optional **Mixnet and Tallier** may be required if the authentication protocol is incorporated into a voting scheme based on mix-nets, described in section 5.6.

### 5.3.4 Initial Data Stored on Key Entities

We now identify the initial data we assume to be stored within the key entities. With reference to Fig 5.3, we assume that the self signed $Cert_{VA}$ can be validated by keys securely delivered out of band. We assume that $VA$ signs $Cert_{VT}$, containing $VT_{PB}$, and $Cert_{IA}$ containing $IA_{PB}$. And we assume that $IA$ signs $Cert_{VC}$ for each $VC$

containing a unique $VC_{PB}$.

We further assume that a voter registration process takes place during which the voter's register is updated with eligible voters. As discussed in Sections 5.1.1 and 5.1.2, a national eID registration program exists where citizens are issued multi app smart-cards that contains their biometric details, cryptographic keys and an electronic voting application.

We thus assume that the following data is stored on each entity:

### Stored at Voting Authority

Public key certificate of Voting Authority $Cert_{VA}$

Private key of Voting Authority $VA_{PV}$.

### Stored at Issuing Authority

Public key certificate of the Voting Authority $Cert_{VA}$

Public key certificate of the Issuing Authority

Public key certificate of Issuing Authority $Cert_{IA}$

Private key of Issuing Authority $IA_{PV}$

Symmetric key shared between the Issuing Authority and Voter's Card $K_{IA,VC}$.

Voter's Identity $ID_{VC}$ and corresponding Voter's registration data.

### Stored on Voter's Card

Public key certificate of Voting Authority $Cert_{VA}$

Issuing Authority's Public key certificate $Cert_{IA}$

Public key certificate of the Voter's card $Cert_{VC}$

Private key of the Voter's card $VC_{PV}$

Symmetric key shared between Issuing Authority and Voter's Card $K_{IA,VC}$

$BiometricTemplate$ of the Voter

Voter's Identification $ID_{VC}$. $ID_{VC}$ may optionally be used by $VT$ to identify $VC$'s expiry date and geographical region where $VC$ is valid.

**Stored on Voting Terminal**

Voting Authority's certificate $Cert_{VA}$

Identity of Voting Terminal $ID_{VT}$

Public key certificate of Voting Terminal $Cert_{VT}$

Private key of Voting Terminal $VT_{PV}$.

$ID_{VT}$ may optionally be used by $VC$ to identify geographical location of $VT$.

**Stored on VTerm**

No data used in the protocols (e.g. keys, certificates, IDs) is stored on $VTerm$.

### 5.3.5 Protocol Objectives

The primary objective of this protocol is to provide Eligibility Verifiability. We wish to prevent impersonation of abstained voters and prevent double voting by legitimate voters. To this end, we define the following goals:

**G-1** Mutual authentication between $VC$ and $VT$ to prevent use of illegitimate cards and rogue terminals.

**G-2** Verification of Voter by $VC$ to reduce trust placed on poll workers to verify voter's identity.

**G-3** Mutual authentication between $VC$ and $IA$, to ensure that $IA$ is communicating with a legitimate $VC$ and vice versa.

**G-4** Authenticated messages to prevent $IA$ or any other entity from including votes for abstained voters and to prevent replay attacks.

### 5.3.6 Attackers Goal

The attackers intention is to change the outcome of the election, we state below some of the ways the attacker intends to achieve this.

**AG-1** Attacker's intention is to impersonate the voter by using an illegitimate card not issued by $IA$.

**AG-2** An attacker acting as a rogue terminal can deceive a legitimate $VC$ and steal sensitive card information. A rogue terminal can also disenfranchise a voter by making the voter believe she has cast a legitimate vote

**AG-3** The goal of the attacker is to allow an impersonator get authenticated by presenting a legitimate $VC$ on behalf of an abstained voter

**AG-4** Attacker's goal is to include ballots for abstained voters without being noticed. This may include impersonating deceased citizens.

### 5.3.7 Assumption of Voting Scheme

In this section some assumptions are listed about the mutual authentication voting scheme

#### Functionality Assumptions

We make some assumptions on the functionalities of different entities:

**A-1** We assume the existence of a Public Key Infrastructure (PKI), based on National Electronic ID schemes already in used in some countries. An example is the in I-voting in Estonia [78].

**A-2** We assume there is only one $VA$.

**A-3** VA is able to install a self-signed public key certificate $Cert_{VA}$ on each $VT$

**A-4** IA has a public key certificate $Cert_{IA}$, signed by $VA_{PV}$ which is installed on $VC$.

**A-5** $IA_{PV}$ does not leave $IA$.

**A-6** $IA$ stores the electoral roll and can check which voters have voted. This will be used later, but not needed for initialization.

**A-7** Each $VTerm$ has a Rich Execution Environment contains a trustworthy $VT$ which is a Trusted Execution Environment as discussed in Chapter 6.4.2. Each $VTerm$ is also equipped with a pin pad, card reader and biometric capturing device

**A-8** We assume $VT$ is bound physically or cryptographically to $VTerm$.

**A-9** Each $VT$ is able to carry out cryptographic functions.

**A-10** Each $VC$ is able to carry out cryptographic functions on behalf of the Voter. The $VC$ will be Multi App smartcard already issued to voters for eID.

#### Trust Assumptions

**A-11** We assume $VC$, $VA$, $IA$, $VT$ are trustworthy.

**A-12** We do not trust $VTerm$. However, we assume the voting terminal supports Global Platform Trusted Execution Environment (TEE) as discussed in Chapter 3, thus $VTerm$ is the REE and $VT$ the TEE. We assume that based on Global Platform TEE architecture, using the Trusted User Interface (see Section 3.3.2),

a secure session can be set up between $VT$ and screen and keypads when ballot forms are being filled.

**Data Transport, Generation and Storage Assumptions**

**A-13** We assume a secure mechanism exists for installing $CertVA$ on $VT$

**A-14** We assume a secure mechanism exists for installing $CertVA$ and $CertIA$ on $VC$.

**A-15** We assume a secure channel exists between $VT$ and $VA$, and $IA$ and $VA$ in order to deliver unsigned (or self-signed) public keys for signing.

**A-16** We assume $VC$ can generate a public/private key pair, or that $IA$ installs these when issuing the card and subsequently destroys $VC_{PV}$. $VC_{PV}$ never leaves $VC$

**A-17** We assume $VT$ can generate a public/private key pair and $VT_{PV}$ never leaves $VT$

**A-18** We assume $VA_{PV}$, $IA_{PV}$ and $VT_{PV}$ are only used for signing.

**A-19** We assume that cryptographic keys based on best practice [28] and currently supported by Multi App Smartcards, as discussed in Section5.3.1 are used in our protocols. For symmetric encryption AES 256, for asymmetric encryption RSA 2048 or 4096 bits or ECC keys 224, 256 or 384.

## 5.4 Mutual Authentication Protocol Run

In this section we show how the protocol runs. All the entities in the protocol know what messages to send, receive and how to respond in each stage of the protocol.

### 5.4.1 Voter Registration

Before we describe the various phases of the protocol run, the voter needs to be registered to the constituencies they are allowed to vote in. Since we are building this scheme

using existing infrastructures, we assume that the voter is issued a Multi Application National Electronic ID card which can be used for voting. This eID is a smartcard with cryptographic and biometric match-on-card capabilities. For simplicity we refer to the eID as the Voter's Card ($VC$). $VC$ is an instance of the voter and can carry out cryptographic operations on behalf of the voter.$VC$ also has the voter's biometric details stored in it.

We assume that $VC$ is only issued to eligible voters and it knows the councils voters are eligible to vote.

### 5.4.2 Protocol Overview

This protocol is divided into the following four phases which broadly reflect our objectives.

**Initialization:** Exchange of messages to start authentication phase.

**Card and Terminal Authentication:** Mutual authentication between $VC$ and $VT$.

**Voter Verification:** Assurance that $VC$ is being presented by a legitimate voter. In this phase the Voter ($V$) will enter a $PIN$ and $BiometricData$ via $VTerm$.

1. Biometric Authentication- Biometric authentication using a smart card is a two-factor authentication in which the voter has a smartcard (something you have) and physiological biometric features of the voter (something you are). In this protocol, matching of the voter's biometric features would be done on the card using a biometric template stored on the card in a process called match-on-card. The fact that the template is stored on the card means it cannot be modified or extracted. Biometric authentication gives a high level of assurance that the voter is who he says he is and he is live during the transaction.
2. Pin Verification- The voter puts in his pin using a pin pad at the Voting Terminal. $VTerm$ sends this PIN to the Voter's card, $VC$ encrypts this

pin using a pin encryption key and send this to the issuer via the terminal. The Issuer decrypts the pin, verifies this pin, and sends a message back to the $VTerm$ that the pin has been verified. In reality, this encrypted PIN is sent along with other data sent by the card to the Issuing Authority during Card and Issuing Authority authentication. Online pin verification have been proposed for EMV transactions [104].

However in our generic scheme, Pin verification is an optional mode of cardholder verification and its use would depend on the scheme our authentication protocol is incorporated into and environment it is deployed

**Card and Issuing Authority Authentication** In this phase of the protocol, $VC$ and $IA$ mutually authenticate. Vote casting also takes place during this phase

### 5.4.3 Notation and Definitions

| | |
|---|---|
| $ID_X$ | Entity X's identifier. |
| $X_{PV}$ | Entity X's private key. |
| $X_{PB}$ | Entity X's public key. |
| $Cert_X$ | Entity X's public key certificate. |
| $K_{X,Y}$ | Secret key shared between X and Y. |
| $N_X$ | Nonce generated by entity X. |
| $(M)_K$ | Encryption of message M with key K. |
| $MAC_K(M)$ | Message Authentication Code on M with key K. |
| $S_X(M)$ | Sign message M with X's signing key. |
| $X\|\|Y$ | Concatenation of X and Y. |

### 5.4.4 Protocol Steps

The protocol run begins when Voter $V$ inserts $VC$ into $VTerm$. $VT$ then sends $VT_{ID}$ together with $Cert_{VT}$ and a random nonce $N_{VT}$ to $VC$ in message **M-1**.

**M-1** VT $\rightarrow$ VC: $Cert_{VT} \parallel ID_{VT} \parallel N_{VT}$

**M-2** VC $\rightarrow$ VT: $S_{VC_{PV}}(ID_{VC} \parallel N_{VC} \parallel N_{VT}) \parallel (ID_{VC} \parallel N_{VC} \parallel K_{VC,VT})_{VT_{PB}} \parallel Cert_{IA} \parallel Cert_{VC}$

**M-3** VT $\rightarrow$ VC: $S_{VT_{PV}}(PIN \parallel BiometricData \parallel N_{VC}) \parallel (PIN \parallel BiometricData)_{K_{VC,VT}}$

**M-4** VC $\rightarrow$ IA: $MAC_{K_1}(ID_{VC} \parallel N'_{VC} \parallel BiometricVer \parallel PinVer) \parallel (ID_{VC} \parallel N'_{VC} \parallel PIN)_{K_2}$

**M-5** IA $\rightarrow$ VC: $MAC_{K_1}(ID_{VC} \parallel N'_{VC} \parallel N_{IA}) \parallel N_{IA}$

### 5.4.5 Protocol Description

The protocol steps and message processing are explained in the following, and illustrated in Fig. 5.4.

#### Card and Terminal Authentication

On receipt of message **M-1**, $VC$ extracts $VA_{PB}$ from the stored $Cert_{VA}$ and uses this to verify $Cert_{VT}$. If $Cert_{VT}$ cannot be verified, then the protocol stops.

$VC$ then generates a fresh nonce $N_{VC}$ and a session key to be shared with $VT$ ($K_{VC,VT}$).

$VC$ then signs $ID_{VC}$, $N_{VC}$ and $N_{VT}$. $VC$ also encrypts $ID_{VC}$, $N_{VC}$ and $K_{VC,VT}$, with $VT_{PB}$ retrieved from $Cert_{VT}$ and sends all of this along with $Cert_{VC}$ and $Cert_{IA}$ to $VT$ in **M-2**.

$VT$ then validates the certificate chain and, assuming correct validation, retrieves $IA_{PB}$ and $VC_{PB}$ from the certificates. $VT$ can then validate $ID_{VC}$ and the signature on $N_{VT}$ sent in message **M-1** thus authenticating $VC$.

If this check fails the protocol is aborted and $VC$ is ejected from $VTerm$.

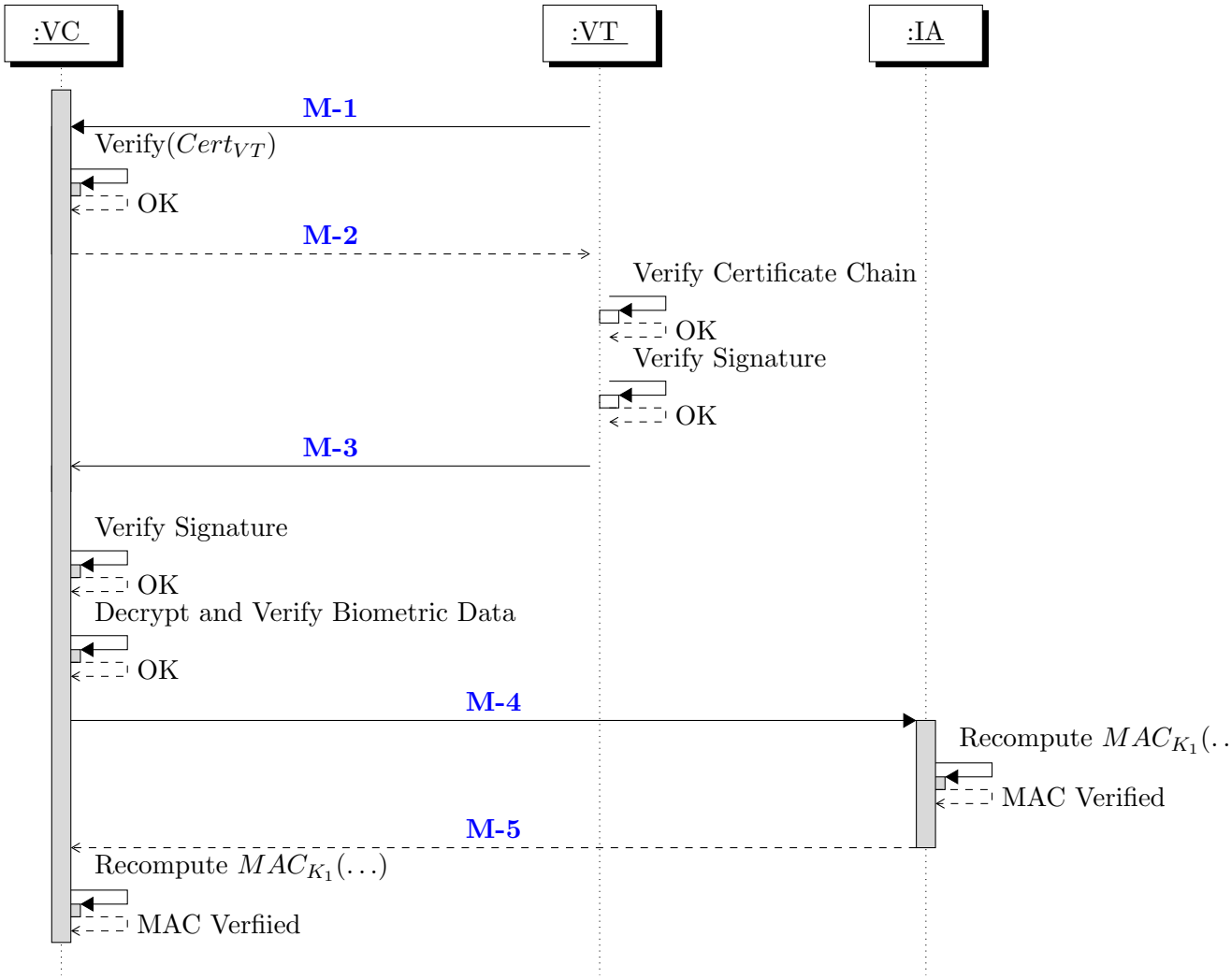


Figure 5.4: Mutual Authentication Protocol

At this point $VT$ can sign $N_{VC}$ and return this to $VC$ in message **M-3**, $S_{VT_{PV}}(\ldots N_{VC})$. Thus, upon receipt of message **M-3**, $VC$ can authenticate $VT$.

### Voter Verification

Our scheme allows the use of multi factor entity authentication. Although possession of $VC$ and verification of biometric data may be sufficient, an optional a third factor (PIN) may also be used.

Prior to sending **M-3**, $VTerm$ prompts the voter to input biometric data and an optional PIN. $VT$ then signs the PIN, biometric data, and the nonce, $N_{VC}$, received in **M-2** and used for mutual authentication above.

$VT$ also encrypts the biometric data, PIN and $N_{VC}$, using the session key $K_{VC,VT}$ and send this to $VC$.

On receipt of **M-3**, $VC$ validates the signature and decrypts the encrypted message. $VC$ then validates the biometric data received against the $BiometricTemplate$ stored on $VC$.

$VC$ knows **M-3** is fresh because it contains $N_{VC}$ sent in **M-2**, also the message was encrypted using the session key shared between $VC$ and $VT$. If this check fails the process is aborted, if not the protocol progresses to the next phase.

$VC$ verifies the biometric data and, if present, the PIN. $VC$ may optionally encrypt the PIN using $K_{VC,IA}$ and send this to $IA$ for further verification. Other on-line PIN verification methods such as those proposed in [105] could be adopted here.

### Card and Issuing Authority Authentication

After verification of biometric data, $VC$ has assurance that the card was presented by a legitimate voter. $VC$ then computes a MAC using a key, $K_1$, derived from $K_{VC,IA}$. The MAC is computed over a new nonce, $N'_{VC}$, a 'Biometric Verified' tag, an optional 'PIN Verified' tag, and $ID_{VC}$. $VC$ then sends this to $IA$ in **M-4**.

On receipt of **M-4**, $IA$ computes a MAC over same set of data to verify it, thus authenticating the data received from $VC$ since possession of $K_{VC,IA}$ has been demonstrated. This data allows $IA$ to carry out further checks on voter eligibility if required.

$IA$ then generates a generates nonce $N_{IA}$, and computes a MAC over $ID_{VC}$, $N'_{VC}$ and $N_{IA}$ using $K_1$. $IA$ then sends this MAC along with $N_{IA}$ to $VC$ in **M-5**. When $VC$ receives this message, $VC$ computes a MAC over $ID_{VC}$, $N_{IA}$, and $N'_{VC}$. If the MAC is verified correctly $IA$ has been authenticated by $VC$.

After authentication has been completed, the voter can then proceed to cast his vote. How exactly this happens would depend on the scheme this authentication protocol is incorporated into.

## 5.5 Analysis of Mutual Authentication Protocol

In section 5.2.2, we identified a number of security requirements for an e-voting scheme in an untrustworthy environment. We also defined, in section 5.3.5, a number of goals that needs to be met to provide Eligibility Verifiability in this environment. We now present an argument to show how these goals are met by our protocol, and more general security requirements satisfied by integrating this protocol into a mixnet based e-voting scheme.

### 5.5.1 Mutual Authentication Protocol

We now present a descriptive argument for each goal identified in section 5.3.5, showing how it is met by our protocol.

**G-1 Mutual authentication between $VC$ and $VT$ to prevent use of illegitimate cards and rogue terminals.**

Both $VC$ and $VT$ generate nonces which they include in the signed messages exchanged during the authentication phase. $VC$ can verify the message **M-1** came from $VT$ by using $VT$'s public key to verify $VT$'s signature. This key is retrieved by $VC$ from $Cert_{VT}$ signed by $VA$, a trusted entity. Likewise, the $VT$ retrieves $VC$'s public key from $Cert_{VC}$ signed by the $IA$ and can verify this key is correct by verifying the $IA$'s public key certificate signed by $VA$. $Cert_{VA}$ is stored on the terminal from which $VT$ can retrieve $VA$'s public key.

For a rogue terminal or MitM to fool $VC$, the MitM would have to be in possession of a legitimate $VT$'s private key which we assume is infeasible to retrieve since this key is securely stored in a tamper resistant module i.e. HSM; and messages are signed using this key, including the terminal generated nonce, to prevent replay of recorded messages from an old session.

Similarly, an adversary cannot act as a legitimate (attack goal **AG-1**) $VC$ without knowledge of $VC$'s private key, which we assume never leaves the tamper resistant device. Moreover, messages are signed using $VC$'s private key and a nonce is included to prevent replay of old messages.

**G-2 Voter verification by the Voter's Card ($VC$) to reduce trust placed on poll workers.**

Our protocol offers multi factor authentication: possession of $VC$, knowledge of PIN and biometric data. An adversary cannot vote on behalf of an abstained voter even with possession of $VC$ (attack goal **AG-1**) because he still has to present biometric data for on-card matching verification with a biometric template. The template is stored on $VC$ and never leaves the card and the biometric capture device is assumed to have liveness checking capability. $VT$ signs the voter's biometric feature it captures,

PIN, and card generated nonce and sends them back to the card for verification. So an adversary or MitM cannot replay old biometric details because the card would notice that the nonce is not the same as that it generated and abort the process. This prevents anyone from voting on-behalf of an abstained voter even if the voter connives and gives his card to an impersonator.

**G-3 Mutual entity authentication between the $VC$ and $IA$, to ensure that issuer is communicating with a legitimate Voter's card and vice versa**

$VC$ and $IA$ share a symmetric key which is used to compute a MAC over the data sent to $IA$ in **M-4**. The shared key $K_{VC,IA}$ used to compute the MAC key, never leave $VC$. An adversary impersonating $IA$ cannot generate a MAC and send it to $VC$ because the adversary doesn't know $K_{VC,IA}$. If an adversary decided to replay old messages both the $IA$ and $VC$ would notice because these would contain a nonce different from that which was generated during this current protocol run and abort the process. The fact that an attacker cannot compute a legitimate MAC without knowledge of the shared symmetric key implies that a rogue Issuing Authority cannot fool a voter into authenticating to a wrong entity and vice versa.

**G-4 Authenticated messages to prevent $IA$ or any other entity from including messages (votes) for abstained voters and to prevent replay attacks.**

Messages exchanged between $VT$ and $VC$ are digitally signed and verified by both parties giving the messages data origin authentication. Encrypted ballots are signed by $VC$ and $IA$ to give data origin authentication and posted on the web bulletin board for everyone to verify. For $IA$ to impersonate $VC$, it needs to have access to $VC$'s private key which is stored securely on $VC$ and never leaves $VC$. Note that even if an entity is authenticated that doesn't necessary guarantee the origin of the message but including digital signature gives us data origin authentication, nonces gives the message freshness and these prevent corrupt authorities from casting votes, replaying old ballots

on behalf of abstained voters without having knowledge of the securely stored voter's private key.

### 5.5.2 Formal Analysis

Our protocol was subjected to mechanical formal verification using the Scyther formal verification tool [61] and was modelled using the formal modeling of security protocols and their properties defined in [60]. The scyther script is presented in Appendix A.

The adversarial model used is the network threat model by Dolev Yao [66]. In the Dolev-Yao model [66], the attacker has complete control over the communication network. The attacker is active and can intercept messages and insert any message, as long as he is able to construct its contents from his knowledge.

In Scyther, each entity in our protocol is defined using a *role*, and an event describes *send* and *recv* (receive) messages between roles (entities). A *claim* event specifies the goals of the protocol that requires verification. Using Scyther we show that secret keys, biometric data and PINs remain secret in the face of an active attacker. We also show that $VC$ and $IA$ are alive during the protocol run and are mutually authenticated using a MAC over an agreed set of data, and this message cannot be formed by a MitM because symmetric key shared between $VC$ and $IA$ remain a secret and cannot be obtained by monitoring or even analysing the communication between these entities.

## 5.6 Protocol Integrated with Mixnet Schemes

Our generic mutual authentication scheme could be an add-on to existing e-voting schemes such as blind signature schemes and efficient mixnet schemes. This would provide voter eligibility, ballot authentication to prevent voting for abstained voters and to prevent ballot stuffing.

In this section we focus on integrating our protocol into a mixnet and vote tallier. A mix-net is a group of mix-servers that take a list of encrypted values as input from a

sender, randomises this and produces an output that cannot be linked to the sender but still corresponds to the input list even though the individual link between the input list and output is hidden. Different types of mix-nets have been proposed in the literature but in our scheme we use the re-encryption mixnet with efficient proof.

In an e-voting scheme, the re-encryption mixnet takes as an input a set of encrypted ballots, re-encrypts these ballots and produces encrypted ballots as output. After this the ballots can be decrypted separately. For an efficient proof, each mixnet server produces a proof of correct operation showing no values have been altered or added during the shuffle it did. These proofs can be verified publicly to confirm correct shuffle and a single alteration would make the proof fail.

In the following we show how we can integrate a re-encryption mixnet into our mutual authentication protocol. We add a Tallying Authority ($TA$) to the existing authorities in our generic model of Fig 5.3, and an 'append only' web bulletin board ($WBB$) where information such as encrypted ballots are posted for verification purposes. We assume that the tallier generates a threshold public/private key pair with the public key signed by $VA$ and contained in a certificate $Cert_{TA}$, stored on $VT$ while the private key shares between talliers are kept secret and would require at least t-out-of-n talliers to decrypt a ballot.

### 5.6.1 Modified Protocol Steps

Integrating a re-encryption mixnet into our scheme, the protocol messages **M-1** and **M-2** remain the same. We include some extra data in **M-3**, **M-4** and **M-5** which represent voter verification and mutual authentication between $VC$ and $IA$ which we show below.

**M-3′** VT $\rightarrow$ VC: $S_{VT_{PV}}(PIN \parallel BiometricData \parallel FilledBallot \parallel N_{VC}) \parallel (PIN \parallel BiometricData \parallel FilledBallot)_{K_{VC,VT}} \parallel Cert_{TA}$

**M-4′** VC → IA: $MAC_{K_1}(ID_{VC} \parallel N'_{VC} \parallel BiometricVer \parallel PinVer) \parallel (ID_{VC} \parallel N'_{VC} \parallel PIN)_{K_2} \parallel S_{VC}(Ballot_i \parallel (K_3)_{TA_{PB}})$

**M-5′** IA → VC: $MAC_{K_1}(ID_{VC} \parallel N'_{VC} \parallel N_{IA}) \parallel N_{IA} \parallel S_{IA_{PV}}(Ballot_i \parallel (K_3)_{TA_{PB}})$

### 5.6.2 Modified Protocol Description

After mutual authentication between $VC$ and $VT$, $VTerm$ prompts the voter to input PIN, biometric data and *fill in a blank ballot*. $VT$ signs all this data, including the $FilledBallot$ encrypts it and sends it to $VC$ in **M-3′**.

On receipt of **M-3′**, $VC$ verifies $VT$'s signature on the message and verifies freshness by checking $N_{VC}$. $VC$ then verifies the biometric data as in the unmodified protocol.

$VC$ then generates a unique temporary key, $K_3$ for encryption of the ballot, encrypts the ballot with this key to produce $(FilledBallot)_{K_3}$ which we rename as $Ballot_i$ to be sent to the tallier.

$VC$ then encrypts $K_3$ using public threshold encryption key of the tallier, $TA_{PB}$. $VC$ retrieves this key from $Cert_{TA}$ sent in message **M-3′**. $VC$ knows this key is correct because it is signed by same key $VA$ used in signing $Cert_{IA}$ stored on $VC$.

$VC$ then sends this to $IA$ in message **M-4′**. $IA$ recomputes and verifies the MAC received in **M-4′** *then checks the electoral roll to see if the voter has already voted.* If the check fails then the process is aborted to prevent double voting. This stage is the same as the Card and Issuer authentication phase in section 5.4, the only difference is the inclusion of the encrypted ballot and temporary key.

$IA$ after authenticating $VC$, computes a MAC over same data as **M-4**. $IA$ then signs $(Ballot_i \parallel (K_3)_{TA_{PB}})$ and sends all of this to $VC$ in **M-5′**.

On receiving message **M-5′**, $VC$ verifies the MAC and can confirm this message comes from $IA$ since the $K_1$ is derived from $K_{VC,IA}$ and the message is not replayed

since it contains fresh nonces generated by $VC$ and $IA$. $VC$ also verifies $IA$'s signature over the encrypted ballot, $Ballot_i$, and $VC$'s temporary encryption key.

$IA$ then posts the following on the bulletin board $BB$ so that it can be publicly verified:

- $S_{IA}(Ballot_i \parallel (K_3)_{TA_{PB}})$
- $S_{VC}(Ballot_i \parallel (K_3)_{TA_{PB}}) \parallel Ballot_i \parallel (K_3)_{TA_{PB}}$

After the mutual authentication between $VC$ and $IA$, $VC$ sends the following to the mixnet server. This data is also posted on the bulletin board:

- $ID_{VC}$
- $S_{IA}(Ballot_i \parallel (K_3)_{TA_{PB}}) \parallel Ballot_i \parallel (K_3)_{TA_{PB}}$

The voter may now choose to cast this ballot or to audit the ballot to see if it is being cast as intended.

If the voter casts the ballot, the voter gets a receipt with all the above information. If he decides to audit the ballot instead, $VT$ sends a vote audit request to the $VC$. $VC$ then decrypts the ballot and presents this information to the voter. The screen displays the plaintext ballot, $K_3$ and other data used in generating the ballot. With this the voter has assurance that $VT$ is acting correctly. This ballot is then discarded and the process started afresh.

To prevent a man-in-the-middle (MitM) from sending a fake audit request to $VC$, $VT$ would sign the nonce generated by $VC$, alongside a freshly generated nonce generated by $VT$. With these nonces $VC$ can verify the audit request is legitimate and fresh because a MitM does not have access to $VT$'s private key and hence cannot generate a valid signature.

After voting has ended, the voter uses the receipt (set of encrypted values) to confirm that his ballot appears on the public bulletin board and hence ballots are recorded-as-cast.

### 5.6.3 Re-encryption Mixnet and Tallying

After the votes are cast, $Ballot_i$ and $(K_3)_{TA_{PB}}$ are sent to the mix-server; we could adopt any of the mixnets used in [103, 132, 62]. The mixnet re-encrypts these values and outputs an encrypted value that cannot be linked to encrypted ballots used as the input. Each mix server produces a proof of correct shuffling that can be audited to confirm. These encrypted values are then decrypted separately upon receipt by $TA$.

Since we use threshold cryptography, a quorum of talliers with their unique private key shares come together to decrypt the encrypted output values. After decryption we are left with $Ballot_i$ encrypted with temporary key $K_3$. The tallier then decrypts the individual votes, tallies and publishes the final result. This process of tallying would be done in a public ceremony witnessed by the various stake holders.

## 5.7 Analysis

In section 5.2.2 we identified a number of security requirements for an e-voting scheme in an untrustworthy environment. We also defined, in section 5.3.5, a number of goals that need to be met in order to provide Eligibility Verifiability in such an environment. We now present an argument to show how these goals may be met by our protocol, and the more general security requirements met by integrating this protocol into a mixnet based e-voting scheme.

### 5.7.1 Protocol Integrated with Mixnet

In this section we provide an analysis of the protocol integrated with the mixnet e-voting scheme. We argue that the combined scheme satisfies the security requirements

described in section 5.2.2.

### Eligibility Verifiability

The voter is authenticated using biometric technology to ensure the voter is present during the transaction. The nonce generated by $VC$ is also included in the signed message containing voter biometric data sent to $VC$ by $VT$ in message **M-3′**. This nonce gives $VC$ the assurance the biometric and PIN information is not being replayed by a MitM from a previous session.

Furthermore, the final ballot signed by both $VC$ and $IA$ is published on the bulletin board for any observer to verify. This means an issuing authority cannot vote on behalf of abstained voters without being caught. $IA$ using the voter's ID and electoral roll can check which voter has voted and prevent double voting.

### Receipt Freeness

The receipt voters get at the end of the election only contains encrypted information, so a coercer cannot get any information from it. It only contains data the voter can look up on a bulletin board at the end of the election to verify that his vote has been recorded-as-cast, thus preventing vote selling and buying. We also assume that since this is a precinct voting scheme, voters would be given some privacy when casting their votes as done in traditional paper based schemes so no one looks over their shoulders.

### Privacy

The voter encrypts his ballot before sending it to the $IA$ to sign it. So $IA$ cannot tell the content of the ballot.

$VC$ sends $ID_{VC}$, $S_{IA_{PV}}(Ballot_i \parallel (K_3)_{TA_{PB}})$ to the mix server. The mix-server re-encrypts the encrypted input list, shuffles it and outputs encrypted values that cannot

be linked to the input, hence maintaining the privacy of voter and ballot secrecy thereby satisfying one of our protocol goals.

The Tallier knows this ballot is legitimate because it was signed by $IA$ but cannot link the key $K_3$ to the voter's Identity since it's a temporary key generated by $VC$ just for encrypting the ballot. Hence this scheme satisfies the privacy requirement of an e-voting scheme.

**Universal Verifiability**

In this scheme, ballots can be audited to provide assurance that $VT$ is acting properly and hence votes are cast-as-intended. A similar approach has been used in other schemes [15, 62]. The voter receives all the relevant information needed to verify his ballot has been recorded-as-cast before deciding on either casting or auditing the ballot. $VT$ has no idea if the voter's decision would be to cast or audit the ballot, so a cheating $VT$ has a high possibility of being caught. The voter using the receipt looks up the bulletin board to verify his votes are recorded-as-cast. The voter has a high assurance the information on the board is correct after previously auditing the ballot to confirm that the voting terminal has acted correctly.

The mix-server provides a proof of correct shuffle, the tallier requires a quorum of talliers for decryption to retrieve the voter's temporary key. The ballots are then decrypted and counted in a public ceremony witnessed by party representatives and observers. This gives assurance that the vote has been counted-as-recorded. The voter's encryption key reaches the end of its key life at the end of every election since it is a temporary key, so it cannot be reused. Standard key management principles used for managing keys at the end of their lifecycle should be deployed in dealing with these keys.

## 5.8 Discussion

Our aim, with this work, was to reduce the trust placed on voters, electoral officials, and voting terminals used in precinct e-voting schemes. The protocol presented in section 5.4 migrates this trust to a number of trusted entities as illustrated in Fig 5.3. Thus we rely more on trusted technology. We believe this is feasible since the use of trusted computing technology and tamper resistant smart cards and tokens is becoming commonplace.

We do however make a number of assumptions about our voting scheme and while most of the assumptions are reasonable because they are based on existing capabilities of the existing infrastructures such as smartcards.Assumption **A-12** on a voting terminal, may be more difficult to realize, hence we rely on the capabilities of a TEE architecture available in mobile devices such as smartphones and tabs and on personal computers. Trusted Execution Environment on personal computers[12] usually referred to as Trusted Platform Modules (TPMs), can be installed on business PCs.We do not choose a specific ballot type in our protocol because it would depend on the country the scheme is deployed. Nevertheless, code voting ballots used in other schemes, can help to reduce the reliance on assumption **A-12**. With code voting the ballot is generated prior to the elections and distributed to voters using an out of bands channel. A code is generated that represents an encrypted value for the candidate, if this code appears on the Public Web Bulletin Board, then its unlikely the ballot has been altered.

We use biometric technology to give a high assurance that the card is being presented by the voter it was issued to. In both retina and iris biometric technology the accuracy level is rated "very high", which is higher than finger print technology rated as "high". The acceptance level and long term stability of the physiological feature of both iris and retina biometric technology is the same as fingerprint technology and are rated "high". However, the ease of use of iris scan is graded as "medium", "low" for

[12]Business PC TEEs, also called Trusted Platform Modules (TPMs), can be installed on business PCs.

retina scan and "high" for fingerprint technology [1]. The level of security the scheme requires would determine which of these biometric technology would be considered. As discussed in Sections 5.1.1, finger print biometric is already being used to authenticate voters in over 50 countries for elections, the main motivation of our scheme is to build on existing infrastrcuture.

A lot of users are comfortable with Chip and Pin technology used for payments. Furthermore, as discussed in Sections 5.1.2, smartcards with biometric verification and cryptographic capabilities are already being issued to citizens as Electronic Identification Cards to access government resource and make payments. Adopting an e-voting scheme is as much about ease of use and acceptance as it is about technical security. This usability and acceptance reason is part of the motivation to propose a scheme which has a similar authentication concept as Chip and Pin payment technology which is understood to a reasonable level and accepted for use in many countries across the world.

Nevertheless, in e-commerce, anonymity of the card holder is not a required security goal, where as electronic voting requires this. Even though in our scheme, messages are authenticated to give data origin authentication (ballot authentication) it still satisfies anonymity of the voter which underpins privacy of the voter. We have shown in our security analysis section 5.7.1 how we satisfy this property.

The cost of running an election can play a big factor in what sort of technology is adopted. In Estonia, obtaining a National ID cost citizens anywhere from 5 to 50 euros[13] depending on your age, disability and some other factors. In Nigeria, citizens are not charged to obtain national ID card but re-issuance of the card cost 10 dollars [14]. According to a report [81] from the Center for Global Development, Washington DC, the investment needed for enrolment and card issuance for each citizen is about 4-11 dollars. According to a report [80] on the use of biometrics in poor countries for

[13]https://www.politsei.ee/en/instructions/state-fee-amounts
[14]https://nimc.gov.ng/fees/

elections, the cost for election per voters is about 5-20 dollars. Figure 5.5 gives more details on the cost in a few countries.

Since our scheme is based on the existing infrastructure in countries located in **Untrustworthy Environment 1** albeit with some modifications, we do not focus too much on the cost of running these elections.

| Country | Year | Registered voters | Election cost | Biometric technology cost | Per voter election cost | Per voter biometric cost |
|---|---|---|---|---|---|---|
| Benin | 2011 | 4,483,000 | 51,704,000[12] | 12,950,000 | 14.1 | 2.7 |
| Burkina Faso | 2012 | 4,365,000 | 58,000,000 | 23,000,000 | 13.3 | 5.3 |
| Cameroon | 2013 | 5,481,226 | 39,000,000 | 15,000,000 | 7.1 | 2.7 |
| Cote d'Ivoire | 2010 | 5,780,000 | 330,000,000 | 266,000,000[13] | 57.1 | 46.0 |
| DRC | 2011 | 32,000,000 | 360,000,000 | 58,000,000 | 11.3 | 1.8 |
| Ghana | 2012 | 14,031,793 | 124,000,000 | 70,000,000[14] | 8.8 | 5.4 |
| Kenya | 2013 | 14,350,000 | 325,000,000 | 106,200,000 | 22.6 | 7.4 |
| Mali | 2013 | 6,800,000 | 50,000,000 | 14,300,000 | 7.4 | 2.1 |
| Nigeria | 2015 | 70,000,000 | 627,000,000 | Not available | 8.6 | ----- |
| Sierra Leone | 2012 | 2,700,000 | 25,000,000 | 10,000,000[15] | 9.3 | 3.7 |
| Tanzania | 2015 | 23,161,440 | 120,000,000 | 72,000,0000 | 5.2 | 3.1 |
| Zambia | 2011 | 5,167,000 | 67,600,000 | 14,700,000[16] | 13.1 | 2.8 |

Figure 5.5: Estimated Election Cost in Dollars
[80]

## 5.9 Summary

In this section we have proposed a mutual entity authentication protocol based on the existing National Electronic Identification Card already being used in many countries. We have shown that using biometric technology provided by these eIDs we can verify voters eligibility. We have discussed the notion of an untrustworthy environment and how trust assumptions made by many e-voting systems may not hold in real world deployment. We have shown that the use of smartcards, biometrics and authenticated messages can prevent various attacks such as impersonation of voters, polling station officials voting on behalf of abstained voters, vote buying and selling and ballot stuffing

by legitimate voters. We also show how a re-encryption mixnet can be integrated into our generic authentication scheme to achieve the security requirements of an e-voting scheme.

Although we assume eID can be used for electronic voting, we believe it is a reasonable assumption as it already being used in Estonia and provision have been made for it in eIDs issued in Nigeria.

.

# Part III

# Untrustworthy Environment

# II-Remote Voting

# Chapter 6

# TEE Mobile Voting Scheme

*This chapter presents another contribution to the thesis.The internet was not designed with security in mind, this makes it inherently insecure. Any connected device invariably inherits the insecurity associated with the internet. With that in mind a mobile device is considered an untrustworthy environment. The mobile phone is vulnerable to a plethora of attacks which has been widely documented. Nevertheless a mobile device would be particularly useful for an election. Remote voting makes elections more mobile and flexible which is believed would increase voter participation most especially for younger voters who have a low participation rate in elections. The main consideration for mobile voting in this environment is that election violence and voter intimidation discourages legitimate voters that would have otherwise exercised their franchise on Election Day*

## 6.1 Introduction

Electoral violence during the election cycle in certain parts of the world has been widely reported. It discourages legitimate voters from coming out to vote on Election Day because of fear for their safety. Remote electronic voting using mobile devices would

help improve voter's participation in this hostile environment because some voters that would have otherwise abstained can now cast their ballots from the comfort of their homes. However, voting through a mobile device creates a new threat environment that needs to be taken into consideration if it's to be used for a binding election.

A mobile device is considered to be an untrustworthy environment due to all the potential vulnerabilities that could be exploited to carry out various forms of attack, these vulnerabilities have been widely reported. Moreover, mobile devices are connected to the inherently insecure internet, so inherits some of the security issues that exists on the internet.

In this chapter, we propose an electronic voting scheme in a hostile voting environment on an untrustworthy mobile device using group signatures to anonymously authenticate the voter, biometrics to verify the identity of the voter and a Trusted Execution Environment to provide secure execution in an untrustworthy mobile device. First we start off by investigating what constitutes an untrustworthy environment in the next section

### 6.1.1 Contribution

The main contribution is proposing an electronic voting scheme on an untrustworthy mobile device in a hostile voting environment. We consider the hostile physical environment this device would be deployed and the threats that exists based on reported cases of electoral issues from different elections. We then consider the security vulnerabilities that exists on mobile devices, which makes it difficult to guarantee the integrity of mobile voting scheme, this issue was termed the "Secure Platfrom Problem" by Ronald Rivest [82]. With that in mind we proposed our TEE Mobile Voting Scheme, using anonymous group signatures to provide eligibility verifiability of the voter whilst still protecting the anonymity of the voter's identity. The TEE capability is used to provide security within the untrustworthy mobile device.

## 6.2 Untrustworthy Environment II

Chapter 4 and section 5.2, talked about an untrustworthy environment and highlighted reasons why poll-site officials should not be trusted to prevent ineligible voting and ballot stuffing in that environment. In this section, another untrustworthy environment made up of a Hostile Voting Environment and an unsecure Mobile Voting Device for remote voting is presented.

The next section starts by describing a Hostile Voting Environment based on various reports on issues recorded from elections conducted in the real world. The case for the need to vote in this hostile environment using a mobile device is then discussed. However, the mobile device is also considered an untrustworthy environment due to its vulnerabilities and if used for elections could jeopardize the credibility of the outcome. Some of these vulnerabilities are explored in section 6.2.2

### 6.2.1 Hostile Voting Environment

Electoral violence has been widely reported in elections in some parts of the world [32, 72, 40, 54]. It was reported that several members of the opposition party to the incumbent president in the 2008 Zimbabwean elections were either killed or sustained injuries [165]. In Malawi there were reports [127] of small violence, delay in delivering electoral materials to polling units and these issues affected the credibility of the election.

In the 2013 Kenyan elections, instances of violence and intimidation of female candidates were reported although it was less violent than elections conducted in 2007 [167]. In Sri Lanka, election related violence and deaths were reported in the build up to the 2015 elections although not as violent and large-scale as was expected [165] but still impacts on voter's participation and choice of candidates.

Electoral violence have been widely covered and reported in both federal and state elections in Nigeria over the years [32, 71, 143, 54]. In the recently concluded 2019

Nigerian presidential elections, it was reported by various news media that parts of the country witnessed electoral violence, ballot stuffing, snatching of ballot boxes and in a few cases death[1] of voters and electoral officials. In some instances elections were cancelled or rescheduled because of this violence[2].

Understanding how electoral violence works and its effect can help in designing appropriate electronic voting schemes. Electoral violence and electoral fraud has led to the wider conversation of replacing or including a remote voting procedure. Currently Nigeria has a combination of an electronic and a manual system for voting. Voters are issued with a Permanent Voters Card (PVC) that contains voter's biometric detail. On Election Day the voter's eligibility is verified using finger print biometrics and the PVC, after which the process is the same as traditional paper based schemes [25]. Biometric authentication have also been used in many other countries in the world [80, 100, 144, 98] with the aim to reduce illegitimate voting and ballot stuffing

Electoral violence discourages voters that would have otherwise participated in elections from voting come Election Day, impacting on overall voter participation and election credibility [32, 97]. In the 2019 state government elections and local assembly elections, electoral violence resulted in voter apathy and resulted in low turnout[3]. With this in mind we define Election Violence as violence that happens prior to Election Day and on Election Day with the aim to reshape the outcome of the election by discouraging voters from exercising their franchise, destruction of ballots to suppress votes and forceful coercion of voters to vote for candidates of their chosen.

[1] https://www.independent.co.uk/news/world/africa/nigeria-presidential-election-death-toll-violence-latest-lagos-buhari-abubakar-a8794591.html

[2] https://www.reuters.com/article/us-nigeria-election/dozens-killed-in-nigeria-poll-violence-observers-idUSKCN1QD0CV

[3] https://www.aljazeera.com/news/africa/2019/03/voter-apathy-apparent-nigeria-local-elections-190309141520691.html

### 6.2.2 Mobile Devices as an Untrustworthy Environment

Mobile devices have evolved over the years from basic mobile phones for making phone calls and sending text messages to smartphones, PDAs and Tablet with powerful computing ability that could run applications such as video games, banking application, eCommerce application, music streaming applications, identity management application, eGovernment applications etc.

In the mobile device ecosystem, different actors such as operating systems developers, application developers, Mobile network operators, original equipment manufacturers and the users need to co-exist. This gives rise to a complex echo system, were trustworthiness of the mobile device cannot be explicitly guaranteed.

Furthermore, greater connectivity of mobile devices to insecure internet and personal computers exposes them to malware and viruses, even those that affect Traditional PCs. In addition to this connectivity between mobile devices with home television and car radio sets to stream music while driving further creates new security vulnerabilities mobile devices are exposed to.

Mobile devices store security details such as passwords, pins, biometric details and personal information [4], this makes them an attractive target for attackers.

With the volume of applications, of which some are Potential Harmful Applications acting as legitimate applications being developed and deployed to mobile devices, it creates an opportunity for malicious entities to attack mobile devices. Designing an electronic voting scheme for an untrustworthy mobile devices requires mechanisms to prevent or detect ballot alteration. Ronald Rivest [82] coined the term "Secure Platform Problem" to indicate the difficulty in guaranteeing that ballot have not been altered by malware that may be running on the mobile device hosting the voting application, which then makes the devices an untrustworthy environment.

[4] Mobile devices store personal information such as identity details, email details, banking details etc

These threats and attacks that exist in mobile devices have been widely covered [69, 113, 131]. We now discuss some of the threats below, by breaking the threats into the attacker's goals and possible attacker vectors in an electronic voting context.

**Attacker Goals in an Electronic Voting Context**

**AG-1** Steal Information: The goal of the attacker in an electronic voting context is to steal personal information voter registration details and ballots. An attacker also wants to steal security critical details such passwords, pins, biometric details, authentication keys and encryption keys.

**AG-2** Monitor: using malwares such as spyware, the attacker attempts to monitor and record how voters have voted. This is useful to an attacker because it enables them to coerce a voter to cast a ballot that reflects one of their chosen.

**AG-3** Denial-of-service attacks: The goal of the attacker is to disenfranchise a voter, the attacker attempts to achieve this by installing battery draining malware or malwares that could repeatedly switch off devices.

**AG-4** Rooting: the attacker aims to root the device so it has full control of the devices to execute restricted commands and install malware. The ultimate goal of the attacker is to install a vote altering malware or cast ballot on behalf of the voter.

**Attack Vectors**

**AV-1** Stolen Mobile Device: The attacker could steal the device off the voter, access the voting application and cast ballot on behalf of the voter. The attacker could also do a physical side channel attack to steal secret keys

**AV-2** Insecure Public Wi-Fi: with millions of open and insecure public Wi-Fi, it creates an opportunity for an attacker monitoring such networks to steal any information voters send over this network. Attackers could also create a spoof Wi-Fi network

for voters to connect to, which creates an avenue to steal sensitive voting information

**AV-3** Downloading Malicious Software: as mentioned earlier, an attacker can use various means such as social engineering or other stealthy methods to get users to unknowingly download potentially harmful applications onto their devices that could monitor, steal and alter votes

## 6.3 Technical Background

Anonymous group signature schemes forms a pivotal part our proposed voting scheme presented in this chapter. Anonymous group signature technique underpins eligibility verifiability whilst still anonymizing the identity of the voter.

### 6.3.1 Anonymous Group Signature Scheme

Anonymous group signature schemes allows a group member sign a signature anonymously on behalf of a group. The signature verifier can verify that the signature was signed by a valid member of the group without knowing the identity of the signer. This makes anonymous group signature attractive for electronic voting schemes. The BS ISO/IEC 20008-1 standard [3] describes two different anonymous group signature schemes. An anonymous group signature scheme with multiple public keys referred to as a ring signature scheme and an anonymous group signature scheme with one group public key [3, 4]which we would refer to as a group signature scheme. The group signature scheme is based on another scheme proposed in [36].

In the group signature scheme with one group public key, every group member has a unique private key that has a relationship with the group pubic key. The ISO standard's [3, 4] anonymous group signature key with one group public key and a unique private key for each group member, is adopted for the voting scheme proposed in section 6.4.

Unique individual private keys can be generated by the group manager for each group member and distributed to them securely. This would mean that the group manager can generate valid signatures on behalf of the group. Alternatively, each group member can generate their own unique signature taking away the need to trust the group manager not signing messages on behalf of the group. Our voting scheme proposed in section 6.4, adopts the scenario where group members generate their own anonymous signing keys as described in sections 6.5.2.

The group signature process is broken down into different steps but before that is described, some common terminologies from the standard [3, 4] is defined below:

1. Group membership credential: this is a data element specific to the group member generated by the group manager and rendered unforgeable by the group member issuing key.

2. Group membership issuing key: This is a private data element specific to a group manager, this key is also referred to as the issuing key and it is usable only by the group manager in the group issuing process.

3. Group public parameter: these are data elements specific to the group and accessible to all members of the group. The group public parameters are used in generating signature keys

4. Group signature linking base: This is a public data element, specific to a group signature linker, which is involved in linking two signatures in the group signature process

5. Group signature linking key: This is a private data element specific to a group signature linker and usable only by the linker in the group signature linking process. The group manager carries out the function of the group signature linker and the group issuer in the proposed scheme in this chapter.

6. Group public key: This a public data element mathematically related to the group membership issuing key, which is involved in the group membership issuing process. Every group has one group public key and individual private keys generated by each group member.

7. Data Element: this is a bit string, an integer, a set of bit string or a set of integers

### 6.3.2 The Group Signature Process

This sections breaks down the various steps in the group signature process as adopted from the standard [3, 4]

1. Key Generation: the key generation process is an interactive session were data elements are exchanged between the group manager (Issuing Authority) and group member (Issuer SD/TVA on behalf of the voter) to generate the keys. The Group manager generates the public parameters, the group public key, the group membership issuing key which is a private data element needed in generating group member's credentials, the linking key and corresponding linking base.

   The group manager sends the group public key, the group member credentials, the public parameters and linking base to the group member. The group member then generates a member private key which the group manager does not have access to. With the member's private key and group membership credentials the group member generates a group member signature key which is stored securely.

2. Signature Creation: The signature creation process involves the signer creating a signature over a message using the group member signature key. As seen in figure 6.1, the signature creation process takes as input the message, the group

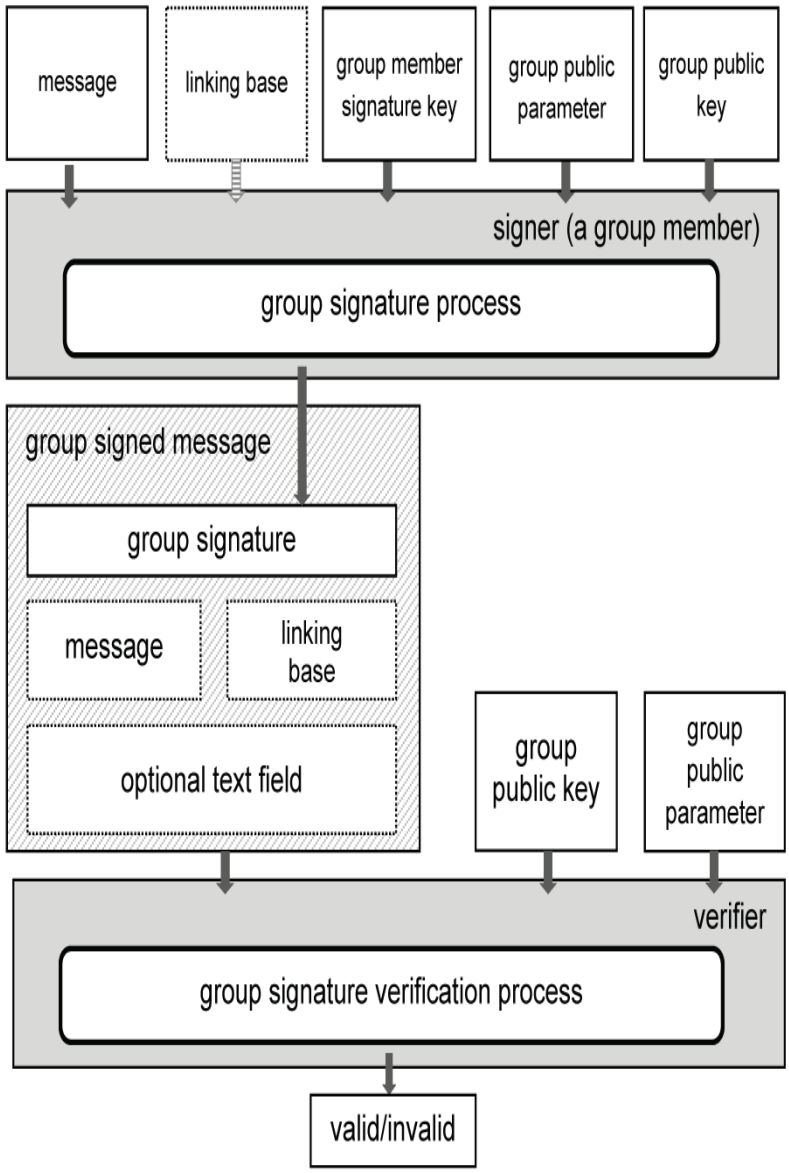


Figure 6.1: Anonymous Group Signature Signing and Verification [3]

member signature key, group public key, group public parameter and the linking base. The linking base enables the signature linker, in this instance the group manager, check if two signatures were signed by the same group member. The output is a group signed message which the group member sends to the group manager/signature verifier.

3. Signature Verification: signature verifier, in this instance the group manager, associates the Group Public Key with group signed message. The group manager inputs the group public parameter, group public key, the group signed message into the group signature verification algorithm. The decision is valid if the group signature is valid or invalid if it was not signed with the valid group member signature key. The group manager also checks if the signer has signed multiple messages using the linking base. In an electronic voting context ability to link multiple signed messages to a particular signer helps to prevent ballot stuffing, a feature which is pivotal to our TEE Mobile Voting Scheme section 6.4

## 6.4 Proposed TEE Mobile Voting Scheme

In 2019, Global Platform announced[5] it shipped 1 billion GlobalPlatform-compliant Trusted Execution Environment (TEE)-enabled processor for the smartphones the previous year. According to a market and consumer data company Statista, smartphone vendors have sold over 1 billion smart phones to end users anually since 2014 [35], see figure 6.2 for more details.

Smartphones are equipped with biometrics authentication capabilities. The iPhone Xs uses facial recognition for authentication, the iPhone 7 uses finger print biometrics [21] and android phones are equipped with various biometric verification capabilities. Some applications on mobile devices require users to be authenticated using biometric technology to access to it or certain security critical aspects. As an example,

[5]https://globalplatform.org/latest-news/1-billion-globalplatform-compliant-tees-shipped-in-2018/

some mobile banking applications require user verification either using biometrics or passcodes before access is granted.

Within the mobile device, there is a Trusted Execution Environment (see chapter 4) where security critical operations is done, secret information stored and keys are generated. We leverage on the security capabilities of the TEE in the TEE Mobile Voting Scheme. Biometric technology is used for voter verification and anonymous group signature for eligibility verifiability.

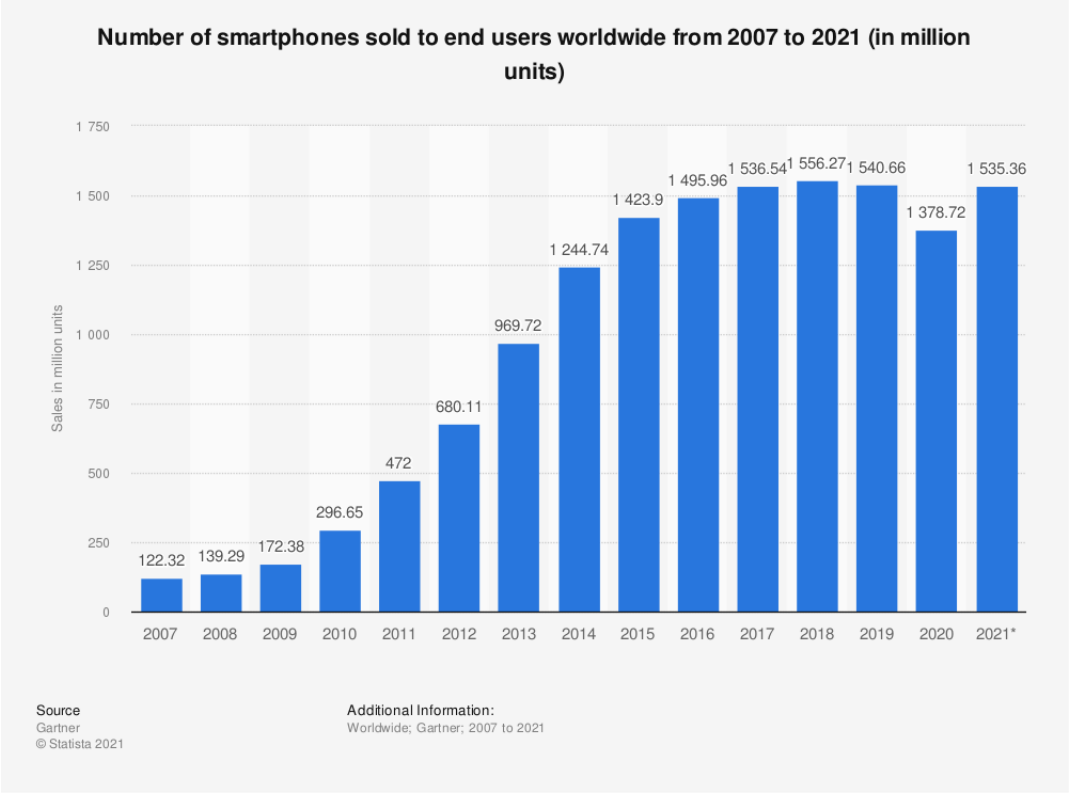


Figure 6.2: No of smartphones sold to end users worldwide from 2007-2021 [35]

### 6.4.1 Key Entities in TEE Mobile Voting Scheme

We describe the various entities involved in this protocol

1. Voter: the individual that owns the mobile device and intends to vote with this

device.

2. The Mobile Device (MD): The mobile device houses the voting application the voter requires to cast his ballot. The Voting Application is installed in the Rich Execution Environment, the Trusted Voting Application and other Trusted Components run in the Trusted Execution Environment as described in TEE architecture in chapter 4. The Rich Execution Environment is Isolated from the Trusted Execution Environment [see section 3.2 allowing security sensitive operation of the Voting Application to be executed in this environment.

3. Election Authority (EA): This entity is in charge of overseeing the elections. We assume this is a trusted authority responsible for certifying keys of the various entities and issuing public key certificates. We assume the Election Authority is the service provider of the TEE and has a Root Security Domain installed on the mobile device for administrative purposes such as creating Security Domains for other entities and Trusted Applications. There is only one central Election Authority and many trusted Issuing Authorities that could be distributed across regions or districts depending on the size of the election

4. Issuing Authority (IA): The Issuing Authority has a Security Domain installed in the TEE on the mobile device. The Issuing Authority has control over this Security Domain. The Issuing Authority registers the voters and has access to the electoral role. The Issuing Authority is also the Group Manager of the group signature scheme, generates the group public key and can verify group signatures signed by the Trusted Voting Application on behalf of the voter.

   In this scheme, the Election Authority is separated from the Issuing Authority, mainly to indicate that they carry out different functions even though the Issuing Authority would normally be an authority under the main Election Authority.

We assume there are many Issuing Authorities responsible for managing anonymous signature groups of different sizes, making our voting scheme more scalable. Putting this into context, as applicable in TPBVS, voters are registered into different districts they are eligible to vote in, managed by the Registration Authority in charge of that district. The same rationale applies with Issuing Authorities in our voting scheme, were IA creates an anonymous signature group and issues membership credentials to eligible voters to join the group.

An Issuing Authority can manage multiple groups and the Election Authority has a central database of all the issuing authorities and groups they manage

5. Tallying Authority (TA): the Tallying Authority is in charge of vote collation. The Tallying Authority has an election private and public key pair. The election encryption public key is stored securely using the TEE Secure Storage on the device. The Tallying authority does not have a Security Domain on the mobile device. The election key is stored securely on the device by the Election Authority and available to the Voting Application for ballot encryption.

6. Voting Application (VA): The Voting Application is installed in the rich execution environment of the device. This application is not considered as secure. The Voting Application is the first point of contact for the voter. The voter fills in his ballot using the Graphical User Interface of the Voting Application displayed on the screen. Just like Mobile Banking Applications, the Voting Application on a device should only be used by the voter is has been registered to. The Voting Application executes any security sensitive operation by connecting with an instance of the Trusted Application running in the TEE.

7. Security Domain (SD): this is an on-device representative of the remote authority. The Security Domain creates Trusted Applications, can generate keys, random numbers and can personalize Trusted Applications with keys. Our scheme has

two Security Domains, Election SD (see section 6.4.5) and Issuer SD (see section 6.5.1).

8. Trusted Voting Application (TVA): this application is the trusted part of the Voting Application where security sensitive operations of the Voting Application such as vote encryption, digital signature and key generation is executed. The Voting Application communicates with an instance of the Trusted Voting Applications using TEE Client API (see chapter 4). The Trusted Voting Application is created by the Issuer Security Domain.

   The Trusted Voting Application uses services provided by other Trusted Applications created by both the Election SD and Issuer SD. As an example, in our TEE mobile Voting scheme, the E-TA is personalized with the election encryption key, if the TVA needs to encrypt a ballot, it uses the ballot encryption service provided by E-TA, E-TA just needs to confirm that TVA has the right authorisation and its within the same Trusted Execution Environment. For simplicity, we assume Issuer SD and TVA are same authority as discussed in our protocol assumptions 6.4.4 and seen in figure 6.5.2

### 6.4.2 Notation and Definitions

| | |
|---|---|
| $ID_X$ | Entity X's identifier |
| $UUID$ | Universal Unique ID of Security Domain |
| $X_{PV}$ | Entity X's private key |
| $X_{Pk}$ | Entity X's public key |
| $X_{GrpPV}$ | Entity X's Group Member Signature Key |
| $NA$ | Nonce generated by IA |
| $NB$ | Nonce generated by Issuer SD |
| $ENC.SK_{X,Y}$ | Secret session key shared between X and Y |
| $MAC.SK_{X,Y}$ | Secret MAC key shared between X and Y |
| $(M)_{ENC.SK_{X,Y}}$ | Encryption of message M |
| $MAC_{SK_{X,Y}}(M)$ | MAC on message M |
| $SIG_{X_{PV}}(M)$ | Sign message M with X's signing key |
| $X\|\|Y$ | Concatenation of X and Y |

### 6.4.3 Protocol Goals TEE Mobile Voting Scheme

We break down our protocol goals into explicit and implicit protocol goals.

**Explicit Protocol goals:**

**G-1** Eligibility Verifiability: In this protocol we intend to verify the identity of the voter, which reduces the chances of voter impersonation or voting by illegitimate voters

**G-2** Ballot authentication: We intend to tie voter's identity to ballots in an anonymous manner to prevent ballot stuffing by legitimate voters whilst preserving voter's anonymity.

**Implicit Protocol Goals:**

**G-3** Verifiability: The voter should verify that his ballot has been cast-as-intended and recorded-as-cast

**G-4** Privacy : Votes should not be linked to voter's identity, meaning no one should be able to tell how a voter voted

### 6.4.4 Assumptions of TEE Mobile Voting Scheme

We now list a set of assumptions about the technology, cryptographic techniques and devices used in our TEE Mobile Voting Scheme.

**A-1** Since a Security Domain can personalise a Trusted Application with keys [5],we assume that a Security Domain that creates a Trusted Application is same as the Trusted Application. Hence Issuer SD is same as Trusted Voting Application since Issuer SD personalizes TVA with the group member signature key. We make this assumption to simplify explanation of the protocol.

**A-2** In this scheme we assume that the Trusted Voting Application has the authority to request for services from any other TA in the scheme. In the GPD Standard [5, 83], the TA providing the service, trusts the other TA requesting for its service by using a trustworthy indicator that guarantees it that both TAs are within the same TEE and any request has not been exposed to the Rich Execution Environment [83]. For example the TVA is responsible for anonymous group signature over the ballot, whilst E-TA is responsible for ballot encryption and another TA responsible for biometric authentication.

For simplicity we assume that the Trusted Voting Application can anonymously sign ballots, encrypt ballots on behalf of a voter and securely carry our biometric authentication of the voter using the appropriate trusted peripherals.

**A-3** We assume the Election Authority stores the Election Encryption key in the Secure Storage of the TEE and TVA can request the services of the Election Trusted Application to encrypt a ballot.

**A-4** We assume that Trusted Applications and Security Domains communicate securely with each other within the TEE using secure communications as specified in the GPD Standard [83]

**A-5** We assume that the TEE enforces strict access control and only grants approved Security Domains/Trusted Applications access to keys

**A-6** We assume no secret keys such as private keys, encryption keys, signature keys ever leave the Trusted Execution Environment

**A-7** We assume a Security Domain that creates another Security Domain or Trusted Application have limited control over it, hence cannot extract or access secret information generated by them.

**A-8** We assume that a Remote Authority cannot extract or access private keys generated by its Security Domain/Trusted Application.

**Functionality Assumptions of Mobile Device**

**A-9** We assume voters own mobile device equipped with TEE capability

**A-10** We assume mobile device has biometric authentication capability

**A-11** We assume every voter has an individual mobile device, with a single voter biometric detail stored in it. This is a reasonable assumption because most mobile devices are owned by single users with one passcode or biometric belonging used in unlocking the device.

**A-12** We assume private key and unique identifier of equipment manufacturer is hardware secured on the mobile device and Election Authority can verify authenticity of device using a hardware Root of Trust and Chain of Trust

we assume the biometric sensor has a liveness mechanism to ensure biometric details of voter being captured is not being replayed.

#### Cryptography Assumptions

**A-13** We do not specify a particular encryption algorithm in our generic scheme, but we assume that a peer of authorities share a private decryption key and a threshold of 75 percent of the peer of authorities need to be available before a ballot can be decrypted. But the exact encryption algorithm or key size will be based on the scheme our protocol is incorporated into

**A-14** For the anonymous group signature keys, we assume keys are chosen based on the ISO/IEC 20008-2:2013 standard [4], which specifies Elliptic Curve Cryptography(ECC) should be used for the cryptographic keys. The standard specifies different ECC key sizes that ranges from 112 bits to 256.

#### Business Assumptions

**A-15** We assume the Election Authority and Equipment Manufacturer/TEE Provider have a business relationship and can use Trusted Execution Environment capability of the mobile device

### 6.4.5 Installing TEE Mobile Voting Scheme Application

The Voter downloads the Election Application (Election APP) from the APP Store on the mobile device or can go to a registration office to have it downloaded and installed on their device. The Trusted Application Manager/Service Provider in this instance, the Election Authority, creates a Root Security Domain on the device when the Election

APP is installed. The Election Security Domain is an on device representative of the Election Authority (EA). Keys and other election relevant data are stored in the Election SD using the TEE Secure Storage.

Before the Election App is installed, Election Authority authenticates the device using a hardware based Root of Trust, to ensure the Election Application is being installed on a legitimate mobile device and not an emulator or cloned device. Election Authority also verifies the authenticity of the Election Application downloaded from the App Store.

After the Election Application and Device authenticity is verified, EA authorises the installation of an Election Trusted Application (E-TA) by the Security Domain using the InstallSD command. Upon proper verification of the Authorisation Command from Election Authority, E-TA is installed in the Trusted Execution Environment. After the installation process of the E-TA, Election Authority personalizes it with the Election Encryption Key needed to encrypt ballot

In this scheme, we assume Election Authority, is in control of the Root Security Domain through which it manages the TEE. We have separated the functions of the Election Authority from the Issuing Authority, the voting scheme has just one Election Authority but many Issuing Authority. (see section 6.4.1)

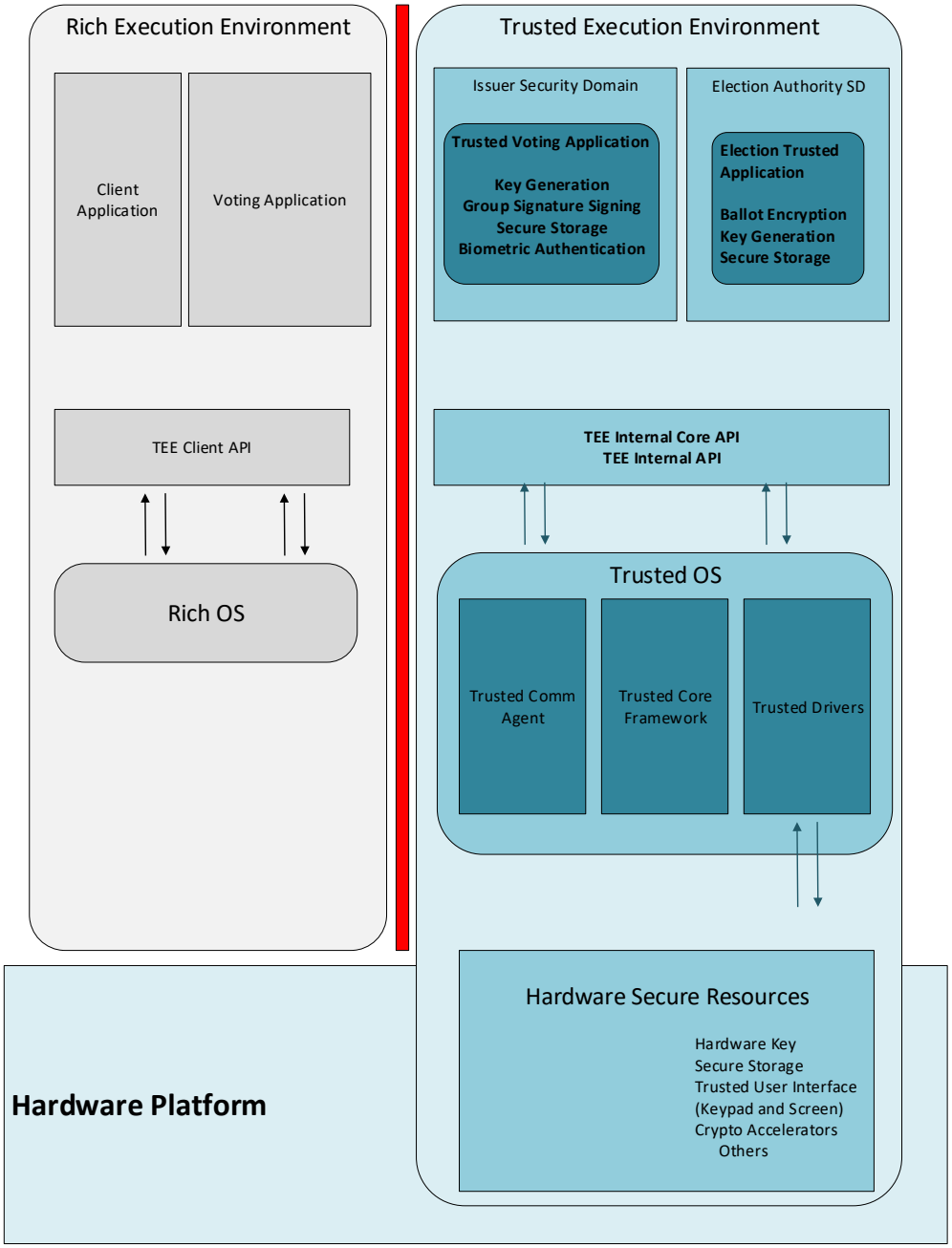


Figure 6.3: TEE architecture with Voting Application Installed

## 6.5 Key Management in Proposed TEE Mobile Voting Scheme

In this section we describe how to create security domains, generate keys, provision keys and create secure communication channels. At the end of the protocol a group member signature key will be generated which will be used to anonymously sign ballots on behalf of the voter as discussed in section 6.6.3.

### 6.5.1 Creating Issuer Security Domain

The Issuer Security Domain creates the Trusted Voting Application, but first the Issuer SD needs to be created. The main goal of the Issuer SD in this protocol is to generate the group member signature key and personalize the Trusted Voting Application with this key. The Group Member Signature is a unique private key used to anonymously sign ballots on behalf of the voter.

We have adopted the process of creating a new Security Domain described in the GPD Stanadard [5]. We now describe how the Issuer Security Domain is created, referring to the protocol description in Figure 6.4.

1. In **Step 1** of the protocol, the Issuing Authority wants to create an Issuer Security Domain in the TEE on the mobile device. It issues an InstallSD command, generates a Universal Unique Identifier (UUID) for the Issuer SD and sends this along with its Public Signature Verification Key $IA_{PK}$ contained in the InstallSD command to Election Authority.

2. In **Step 2**, IA generates an Authorization Token, signs this token using its Private Signing Key and sends all this information in **step 3** in a secure communication session, (see section 3.4.3 for more on Authorization Tokens).

3. On receipt of message 3, Election Authority SD verifies the administrative command to install a new SD from Election Authority and Authorisation Token in

**step 4**

4. Election Authority SD creates a new SD called IA Security Domain using the InstallSD command issued by the Issuing Authority. The InstallSD command provisions an initial Issuer's Public Key $IA_{PK}$ in IA Security Domain (generated in step 1 fig 6.4 ) during installation in **step 5**.Once the IA security domain is created, it stores $IA_{PK}$. Prior to step 5, IA Security domain does not exist, IA Security Domain is only created during **step 5**. IA Security Domain can then use this key to verify any signature from Issuing Authority

5. In **step 6**, the Issuer SD generates an RSA Key pair $IssuerSD_{PK}$ and $IssuerSD_{PV}$ that can later be used for key exchange and stores these keys securely.

6. In **step 7**, Issuer SD sends $IssuerSD_{PK}$ to Election Authority SD

7. In **step 8**, Election Authority SD signs $IssuerSD_{PK}$ and sends this to Election Authority in **step 9**

8. Election Authority verifies EA SD's signature with EA SD's public signature verification key in **step 10**

9. In **step 11**, EA Security Domain returns $IssuerSD_{PK}$ to Issuing Authority. Issuing Authority saves $IssuerSD_{PK}$.

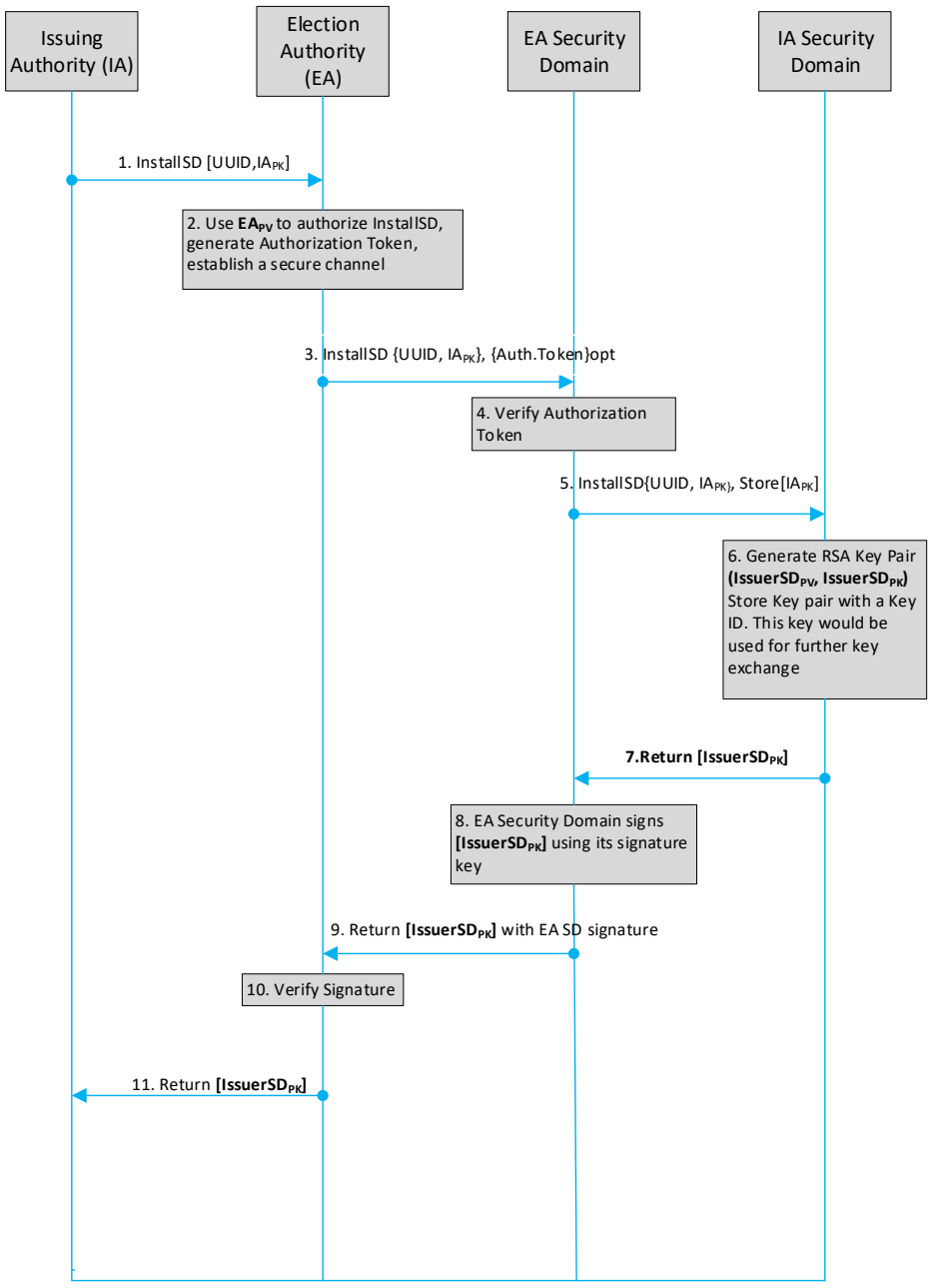


Figure 6.4: Creating Issuer Security Domain

### 6.5.2 Secure Channel Setup and Group Signature Key Generation

We adopt the process of setting up a secure communication channel described in the GPD Standard [7]. We modify the protocol by incorporating an anonymous group signature scheme discussed earlier in section 6.3.1 and section 6.3.2 into it and including extra steps to generate an anonymous group signature by IssuerSD/TVA for the Voter.

IssuerSD/TVA[6] generates the anonymous group member signature key and Issuing Authority doesn't have access to this key, hence cannot sign on behalf of the voter.The anonymous group signature key generated here is used to sign completed ballots as described in the ballot completion phase in section 6.6.3

To set up a secure channel, the Issuing Authority and its Issuer SD mutually authenticate. During the process the issuing authority generates key parameters which both parties use in generating Ephemeral Key Pairs. From the key pairs, they both generate a shared secret, which they use in generating symmetric session keys for Encryption and MACs.

After a secure communication channel is setup between IA and IssuerSD, IA generates and sends group public key, group member credentials, public parameters and linking base to Issuer SD,which IssuerSD uses to generate a group member signature Key.

IA cannot generate group member signature key because it does not know the group member private key generated by Issuer SD corresponding to the Group Public Key. This implies that IA cannot sign on behalf of a voter. Issuer SD then personalizes the Trusted Voting Application with this key, the Trusted Voting Application saves the Group Member Signature Key securely using TEE secure storage and the key never leaves the TEE of the device. We assume that Issuer SD and TVA are the same entity (see section 6.4.4)

[6]as discussed in our protocol assumption IssuerSD and TVA are assumed to be the same entity because after IssuerSD generates the group signature key, it personalizes TVA with that key

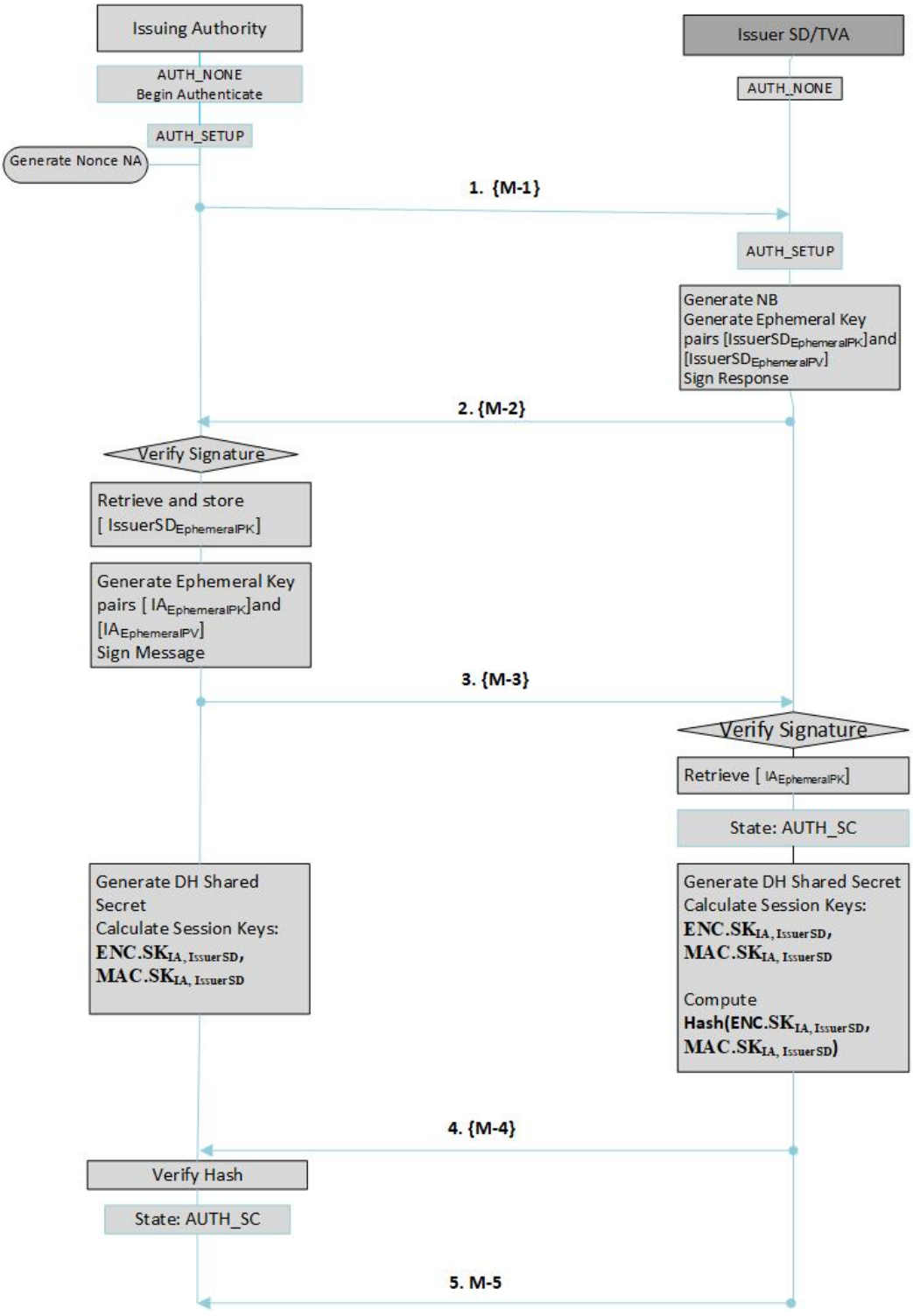


Figure 6.5: Secure Communication Channel Setup

**Protocol Description Group Signature Key Generation**

**M-1** IA $\rightarrow$ IssuerSD: $(ID_{IA} \parallel NA \parallel KeyGenParam \parallel KeyID)$

**M-2** IssuerSD $\rightarrow$ IA: $(UUID \parallel NA \parallel NB \parallel IssuerSD_{EphemeralPK})SIG_{IssuerSD_{PV}}$

**M-3** IA $\rightarrow$ IssuerSD: $(UUID \parallel NA \parallel NB \parallel IssuerSD_{EphemeralPK})SIG_{IssuerSD_{PV}}$

**M-4** IssuerSD $\rightarrow$ IA: $H(ENC.SK_{IA,IssuerSD} \parallel MAC.SK_{IA,IssuerSD})$

**M-5** IA $\rightarrow$ IssuerSD: $(GroupPublicKey \parallel PublicParameters \parallel MembershipCredentials \parallel LinkingBase)ENC.SK_{IA,Issuer_{SD}} \parallel (DATA)MAC_{SK_{IA,IssuerSD}}$

Issuer SD has the Public Key of the Issuing Authority ($IA_{PK}$) installed in it and stored securely during its creation as seen in step 5 figure 6.4. IA also has the public key of Issuer SD received during Issuer SD creation in step 11 figure 6.4

The Protocol describes 3 states, the $AUTH\_NONE$, $AUTH\_SETUP$ and $AUTH\_SC$ state as adopted from the GPD Standard [7].

$IA$ produces the BeginAuthenticate command in the $AUTH\_NONE$ state and transitions into the $AUTH\_SETUP$ state as seen in figure 6.5.

IA generates a Nonce (NA), Key Generation Parameters and a Key ID and sends this as Message 1 to $IssuerSD$, to begin the authentication process. This is reflected in step 1, fig 6.5

When Issuer SD receives **M-1**, with the Key Generation parameters, $IssuerSD$ generates Ephemeral Key pairs, generates fresh nonce ($NB$) and stores its Private Ephemeral Key ($IssuerSD_{EphemeralPV}$). $IssuerSD$ signs its Identity $UUID$, Public Ephemeral Key ($IssuerSD_{EphemeralPK}$), $NA$, $NB$ and sends it to IA in Message 2 (step 2 fig 6.5).

$IA$ receives **M-2**, using the $UUID$ to identify $IssuerSD's$ Public Key, it verifies $IssuerSD's$ signature then retrieves $IssuerSD_{EphemeralPK}$.

$IA$ then generates its own pair of ephemeral keys, saves its Private Ephemeral Key.IA then signs its Public Ephemeral key, $NA$, $NB$ and $UUID$ of $IssuerSD$ and sends this to $IssuerSD$ in **M-3** (see step 3 figure 6.5)

On receipt of **M-3**, Using Key ID received from $IA$ in **M-1**, $IssuerSD$ searches its TEE Secure Storage for $IA_{PK}$ required to verify $IA's$ signature. After $IssuerSD$ successfully verifies IA signature, it retrieves $IA$ ephemeral public key and transitions to the $AUTH\_SC$ state.

The $AUTH\_SC$ state means both parties have been mutual authenticated, secure channel has been set up and keys have been negotiated. With the Ephemeral Public keys exchanged by both parties, using the Diffie-Hellman key exchange protocol $IA$ and $IssuerSD$ generates a Shared Secret from which they can calculate symmetric session keys for Encryption, Message Authentication Codes (MAC), and Hash Encryption Keys.

After session keys have been generated, $IssuerSD$ then responds with a Hash over $ENC.SK_{IA,IssuerSD}$ and $MAC.SK_{IA,IssuerSD}$ in **M-4** (see step 4 figure 6.5). When IA receives this hash from $IssuerSD$, it verifies the hash, after which both parties can then begin to exchange secure messages using a Cryptographic Security Layer[7].

After the secure communication channel is set up, $IA$ generates the group public key, the group membership credentials, public parameters and linking base. IA encrypts all this information and computes a MAC over it using shared symmetric session keys in a symmetric security layer and sends this to $IssuerSD$ in **M-5** (step 5 6.6) and computes a MAC over this data using the MAC key it now shares with $IssuerSD$. $IA$ sends all this information to $IssuerSD$. $IA$ encrypts all this information using symmetric encryption key it now shares with $IssuerSD$

On receipt of **M-5** , $IA$ decrypts the message and verifies the MAC over all the data to check its validity. After confirmation of validity, $IssuerSD$ retrieves the group

[7]The standard [7] describes different security layers, AES in combination with HMAC; AES-GCM [70] and AES-CCM [170] but it allows for other Security Layer Designs

public key, the group membership credentials, public parameters and linking base.

IssuerSD, then generates a group member private key, and with the group membership credential, $IssuerSD$ generates group member signature key. Issuer SD personalizes the Trusted Voting Application with the group signature key and other information received from Issuing Authority securely. Figure 6.3 shows a TEE architecture on a mobile device with the security domains and corresponding Client/Trusted Applications installed for an election

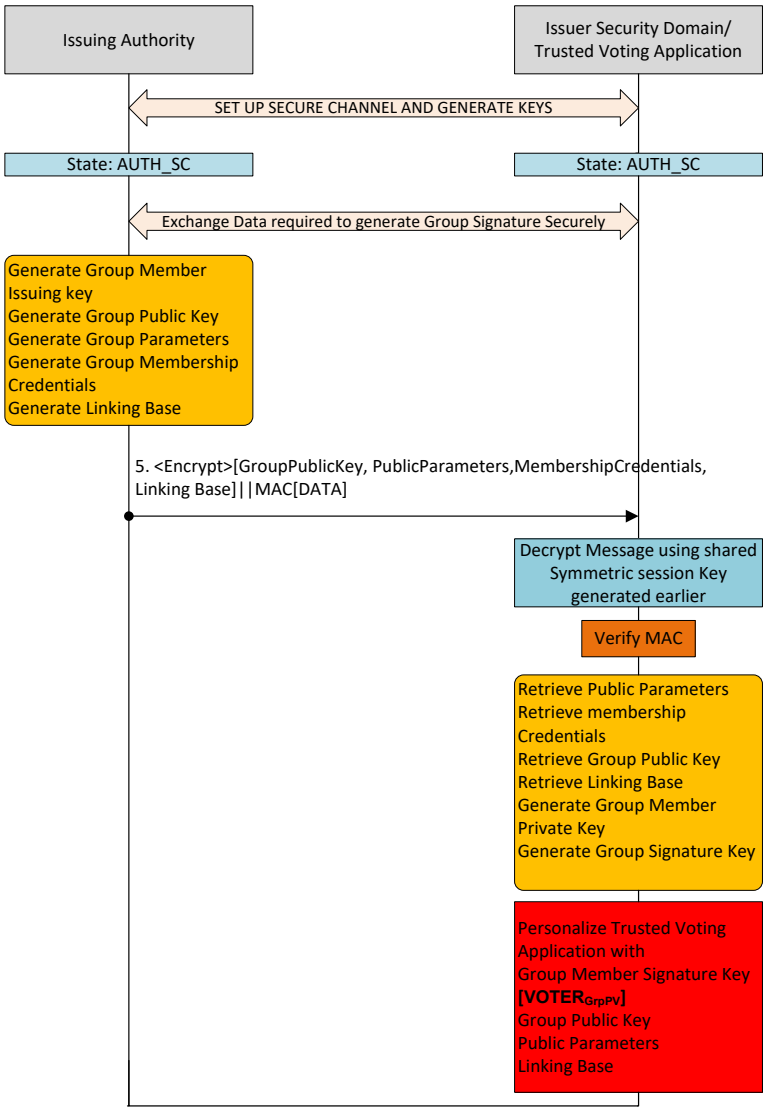


Figure 6.6: Issuer SD Generates Group Member Signature Key

## 6.6 The TEE Mobile Voting Protocol

Using the anonymous group signature algorithm discussed in section 6.3.1, the Trusted Execution Environment Architecture and its security capabilities discussed in Chapter 4, we now break the voting protocol into various stages:

- Step 1: Voter Registration by Issuing Authority. See section 6.6.1
- Step 2: Installation of TEE Mobile Voting Application on Voter's mobile device, described in section 6.4.5
- Step 3: Issuing Authority via Election Authority creates Issuer Security Domain, described in section 6.5.1
- Step 4: Secure communication channel is set up and Issuer SD/TVA generates anonymous group member signature key on behalf of the voter. See section 6.5.2
- Step 5: On election day, voter is authenticated using mobile device biometric technology, before voter can access and complete ballots as described in section 6.6.2
- step 6: Voter completes ballot securely. See section 6.6.3
- Step 7: At the end of the election, ballots are collated and tallied, described in section 6.6.4

### 6.6.1 Voter Registration

Prior to the elections the voter goes to a registration office to get registered. The voter's identity is verified using international passports, national identification card, driver's license or any other accepted forms of identification. The Voting app is then installed on the mobile device of the voter.

Alternatively, the voter can register remotely using any appropriate channel decided by the election authority. The voter downloads the voting app from the app store and

installs the Voting Application as described in section 6.4.5. The voter registers relevant details as specified by the voting scheme and inputs Voter ID received from the Issuing Authority through an alternative channel. The app then prompts the voter to register biometric details, this details are captured and matched with the biometric details stored on the device. This completes the registration phase.

When this data is being captured with Trusted Biometric Peripherals, the Trusted Application sets up a secure Trusted User interface (TUI) session, which implies that the process is isolated and no other application installed in the Rich Execution Environment on the device or malware can interfere with this process.

### 6.6.2 Voter Verification on Election Day

Voter opens his mobile device and then clicks on the Voting Application. The voter logs into the Voting Application located in the REE using his login details. The Trusted Application requests for voter's biometric details in a secure fashion. To achieve this, TA requests for an exclusive TUI, opens up a session in which the Biometric Details of the Voter is captured using the appropriate Trusted Biometric Peripheral without interference from other applications in the REE, unauthorised Trusted Applications in the TEE or Remote Authority.

TA accesses the Biometric Peripherals using the Biometric API. The Biometric Peripherals are under the control of the TEE and located in the Biometric Subsystem [figure 6.7]. With the use of the Biometric API [83] [section 3.3.2] the captured voter biometric detail is matched against voter's biometric template stored securely in the TEE, if this matches successful the Voter's identity is verified and that session is ended. Figure 6.7 shows an example of a generic finger print authentication architecture from the GPD Standard [83]

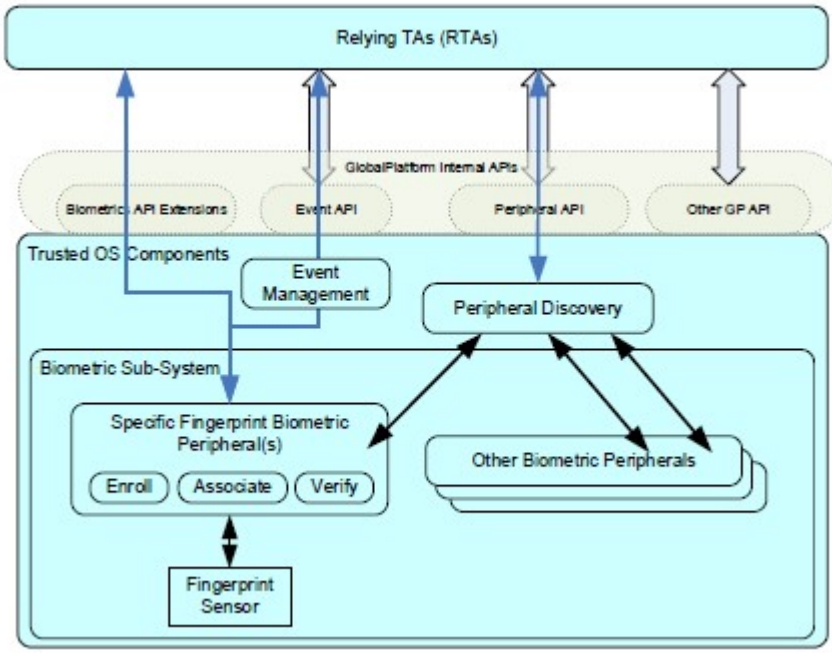


Figure 6.7: Finger Print Biometric Example [83]

### 6.6.3 Ballot Completion

After the voter's identity is verified, the Voter is presented with a blank ballot to fill, the voter fills the ballot using the keypad. All the information displayed and filled by the voter is done taking advantage of the secure display and secure input provided by the TEE TUI. In addition to this TEE TUI prevents multiple sessions being opened while the voter fills in his ballot preventing any interferes with the current session.

After voter fills the ballot and selects the ballot completed option, the filled ballot is encrypted using the election encryption key and signed using the group member signature key. The encrypted and signed ballot is then sent securely to the issuing authority. The authority checks to see if the voter has voted multiple times, with the linking base, the authority can tell if two signatures have been signed by same voter. Depending on the policy, the authority discards the new ballot received, cancels both or uses the more recent ballot if the scheme supports re-voting.

The security indicator feature gives the voter assurance and eventually the Issuing

Authority that the encrypted and signed completed ballot it receives from the Security Domain/Trusted Application on behalf of the voter is exactly what the voter saw and filled. In sections 6.5.2 we have shown how this key can be created in a security domain.

The Issuing Authority then posts the encrypted and signed ballot on the web bulletin board. The voter also has this information saved securely in the secure storage in the TEE and can log into the app to have this securely displayed on the screen anytime, to verify the ballot was recorded as cast. The signatures are then stripped from the ballot and sent to the tallier in a secure manner.

In this scheme, we do not decide which encryption algorithm should be used, that decision will depend on the type of election the scheme is used in. For referendum type elections we suggest using homomorphic encryption algorithm [68, 29] and for write-ins, rank based elections or ballot systems such as Prêt-à-Voter we suggest the use of el-gamal encryption which will be suitable for ballot anonymisation using reencryption mixnet in the tallying phase as seen in [159]

### 6.6.4 Vote Tallying

On receipt of the ballot, assuming the ballots are encrypted using a homomorphic encryption algorithm, the tallier multiplies the encrypted ballots to get the sum of the corresponding plaintexts as the election result. This is done without having to individually decrypt every ballot.

If a mix-net option is opted for, the encrypted ballot is passed through a set of re-encryption mix-net, the output cipher text from the final mix-net server would have no relationship with the input encrypted ballot. The output cipher text, which is the encrypted ballot is then decrypted. The re-encryption mix-net provides a verifiable proof of correct shuffling of the encrypted ballot.

## 6.7 Security Analysis of Group Signature Voting Scheme

A security analysis is presented in this section to show how our TEE Mobile Voting Scheme satisfies the security properties of an electronic voting scheme:

1. Eligibility Verifiability: In the TEE Mobile Voting scheme we use the biometric capabilities of the Mobile Device TEE to capture voter's biometric details and verify this with the template stored on the device. The Biometric technology is part of the Trusted User Interface, which sets up a session and prevents interference from other applications in the REE and unauthorised TAs in the TEE.

   The TEE TUI also guarantees secure displays and secure input which ensure that any request that appears on the screen is exactly what was requested and the relying TA receives the exact voter biometric details captured respectively. Through the Biometric API, the Trusted Application can interact with the Biometric Peripherals to verify the biometric details of the voter is legitimate.

   Prior to the election every eligible voter is registered and eligibility of the voter verified before access to the Application is granted. Every voter belongs to an anonymous signature group, we have described how group signature keys can be established in section 6.3.2. Alternatively the group signature key can be pre-installed. Ballots completed by the Voter is encrypted using the Election Encryption Key and signed using SD/TA's group signature key on behalf of the voter.

   The Issuing Authority passes the signed ballot through the group signature verification algorithm, if the signature is valid then it must have been signed by a legitimate member of the group satisfying protocol goal **G-1**. For an ineligible voter to generate a legitimate signature over an encrypted ballot, it would need to have access to a legitimate group member signature key which is generated within the TEE and never leaves the TEE. In addition to this, to fill out ballot an ineligible voter would need to present his biometric details for verification which

is not achievable if he hasn't been previously registered, still satisfying protocol goal **G-1**.

Voter verification also limits the damage of a stolen mobile device (attack vector **AV-1**) belonging to a legitimate Voter because it is unlikely that an attacker can cast a ballot on behalf of a legitimate voter. This is due to the fact that an attacker would need to pass the biometric check to access the Voting Application. We believe with the liveness checking biometric capability of mobile devices and the TEE TUI that prevents interference from other application on the device it would be very difficult for an attacker to fool the device into believing he is the legitimate vote still satisfying security goal 1. With Apple devices the chances of someone presenting biometric details that matches that stored on the device is 1 in 10,000 for fingerprint biometrics and 1 in 50,000 for Face ID [21]

2. Ballot Stuffing: This scheme prevents voters from voting multiple times satisfying protocol goal **G-2**. This is achieved by anonymously linking ballots to voter's identity without revealing the identity of the voter. The SD/TA signs the encrypted ballot using the group member signature key on behalf of the voter, and sends this to the Issuing Authority. The Issuing Authority verifies the validity of the signature using the group public key. With the linking key, Issuer can tell if any group member has voted multiple times without knowing the identity of the voter. Depending on the election policy, IA can either discard duplicates, accept the last signed ballot if the scheme allows for re-voting as means to prevent voter coercion or revoke the signature key of the voter if the scheme allows for signature revocation

3. Privacy: This scheme satisfies privacy requirements of an electronic voting scheme. The SD/TA encrypts ballot using the election encryption key and this is done in the Trusted Execution Environment of the mobile device.

   The encrypted ballot is then signed using the group member signature key and

sent to the Issuing Authority. IA cannot tell the content of the ballot because IA does not have access to election decryption key satisfying protocol goal 3. IA only verifies the validity of the signature over the encrypted ballot and checks if voter has voted multiple times.

When the ballot is forwarded to the Tallier, the Tallier does not decrypt individual ballots to reveal the content rather it uses the properties of homomorphic encryption that allows the decryption process to multiply the encrypted ballots together to reveal a plaintext that is the sum of the total value of the ballot. Besides the Tallier is not responsible for group signature verification, does not have the group signature linking key, group membership credentials so cannot link a signed ballot to a user.

We note that in very large elections it is not scalable to add up all encrypted ballots together, so we suggest that decryption should be done in batches with Homomorphic encryption, the privacy of the ballot is maintained as the content of each encrypted ballot is not revealed and hence not linked to the voter's identity satisfying security goal **G-4**.

If a reencryption mix-net option is opted for, privacy (security goal **G-4**) is still achieved because the encrypted ballot received by IA from the voter, is sent to a tallier through the re-encryption mix-net. After proper shuffle of the encrypted ballot which involves encrypting a ballot by a mix-net server and re-encryption of the output cipher by the next mix-net server, the final output encrypted ballot has no relationship with the input encrypted ballot.

Furthermore, ballots are filled using the secure input provided by the Trusted User Interface when a session is setup with an instance of the Trusted Voted Application. This implies that malwares, other client applications on the REE and unauthorised SDs/TAs cannot interfere with the session and hence cannot learn what ballot choices the voter has inputted maintaining the Voter's privacy

(security goal **G-4**).

4. Receipt freeness: The Voter doesn't get any physical receipt at the end of the ballot that reveals how he voted. The final encrypted and signed ballot is stored securely in the TEE secure storage. At the end of the election the voter can log into the app, get verified again using biometrics after which he can request for completed ballot. Using the Secure Display capability of the TEE TUI, the encrypted and signed ballot is displayed on the screen, this information is not in plaintext, so the voter hasn't gotten any information he could possibly show to a third party. The Voter can the used this information to verify issuer posted it on a web bulletin board.

5. Verifiability: This scheme does not explicitly achieve individually verifiability, this is implicitly achieved through the secure display and secure input capability of the TEE TUI. The voter sees the security indicator which assures the Voter that ballots displayed on the screen was presented by Trusted Voting Application, hence any ballots filled there in, is secure and cannot be altered by any other application on the device.

   At the end of the election, voters can login into the app to have their encrypted and signed ballot displayed on the screen of the mobile device. The voter can then compare this value with that displayed on the Public Web Bulletin Board. If the value on the screen is same as that on the Web Bulletin Board, then the votes have been recorded-as-cast satisfying verifiability (security goal **G-3**). We say universal verifiability is implicitly achieved because individual verifiability is implicitly achieved by trusting the security capabilities of the TEE technology.

### 6.7.1 Further Security Discussion

In section 6.2.2 we talked about attacker's goal in an electronic voting context. In this section we discuss how these goals are prevented or mitigated

### AG-1 and AG-2

The TEE provides hardware separation in the CPU of a modern mobile device between a trusted operating system and an untrusted operating system which makes it isolated from potential threats from malware running on the untrusted OS in the Rich Execution Environment. This means that if a user downloads a malware onto to a mobile device, its threats are isolated from the Trusted Voting Application that runs in the TEE. All security sensitive operations such as voter authentication, ballot encryption, key generation, group signatures over encrypted ballot is done within the TEE which is isolated from the REE, hence a malware running in the TEE cannot affect these operations.

Furthermore, the TEE Trusted User Interface sets up a secure session between the Client Voting Application and Trusted Voting Application, to offer secure display of ballots and secure input of candidate choice by the voter on the voter's mobile device screen. This secure session prevents interference from any spyware running in the REE that intends to monitor any information the voter inputs on the screen.

If a voter connects to an insecure Public Wi-Fi monitored by an attacker or in control of the attacker, the attacker does not learn any information that could break the privacy of the ballot because all communication between the SD/Trusted Voting Application and Issuing Authority, is after they both set up a secure communication

### AG-3 Denial-of-service

Every voter has a unique set of credentials stored in the Trusted Execution Environment of a TEE. The Issuer SD/Trusted Voting Application sets up a secure communication with the remote Issuing Authority to derive credentials and send completed ballots.

An attacker could steal a mobile device preventing the voter from casting a vote with that device. An Attacker can also set up a fake public Wi-Fi for voters to connect SDs/Trusted Voting Application with remote authority, then drop all communications. The attacker could also, possibly install resource draining malware on device that could

possibly drain battery lives or shutdown mobile devices.

However, we argue that an attacker can only cause voter disenfranchisement on a voter to voter or device to device basis rather than on a large scale. If a voter connects to the Issuing Authority via a Public Wi-Fi in control of the attacker and the attacker decides to drop the ballot, the voter would notice his vote has not been recorded because he wouldn't get any confirmation from IA that his ballot has been received, also he wouldn't have his encrypted ballot stored securely within the TEE on his device. In addition to this, his signed and encrypted ballot wouldn't appear on the public web bulletin which would raise suspicions that would lead the voter to connecting to another network to cast his ballot.

**AG-4 Rooting the device**

An at attack aims to administrative level permissions by Rooting an android device or jail breaking an Apple IOS device. This allows the attacker install Malwares on the mobile device or even install a new operating system on the mobile. Considering our attack vectors in section 6.2.2, an attacker who steals a mobile device can carry out this attack and gain privileged control over the device.

The TEE does a hardware isolation between the Trusted OS and the Untrusted OS running in the REE. This means that the Trusted Voting Applications and all secret credentials are isolated and protected from the untrusted Rich Execution Environment. All cryptographic operations are therefore protected within the TEE, so in a rooted device, a malware cannot access any secret credentials or interfere with any operations within the TEE.

## 6.8 Discussion

In our assumptions in section 6.4.4. we assumed that every mobile device has one owner and one biometric detail registered to that device which is usually the case for

most people who own mobile devices. However, android and apple Devices allow for registration of multiple biometric details, this means that the device may not be able tell which biometric details it is accessing, provided the biometric details presented is a legitimately registered biometric detail on the device, then access to applications is granted.

To address this issue, we suggest that a Unique Universal Identifier called a Voter ID is assigned to every voter during registration at a registration office. The voter ID should be attached to a specific biometric detail when the Voter registers to the Voting Application.To access the Voting Application, the voter first needs to present the Unique Voter ID attached to a specific biometric detail. With this in mind, if a third party registered on the same device, presents a Voter ID that's not tied to his specific Biometric Details then authentication of the third party fails.

Nevertheless, the capability of mobile device registering multiple user biometric details, could allow multiple users the ability to vote with the same mobile device. The GlobalPlatform TEE Framework allows multiple TEEs [83] operate side by side on a device but isolated from each other that makes it possible to have multiple voters registered to a device. With a unique Voter ID attached to each Voter biometric detail, the voting application can tell which particular Voter is being authenticated. We do not cover this possibility in this thesis, but this can be investigated further in future works.

In our protocol assumption, we assumed the biometric sensor has liveness checking mechanism, we believe this a reasonable assumption because with the ITouch [21] on apple devices, before the fingerprint sensor becomes active and the advanced imaging array that scans the finger is triggered, the capacitive ring that surrounds the home button first has to detect the touch of a finger.

As we have discussed earlier a TEE is an embedded platform on a mobile device that offers isolated execution of Trusted Applications away from Client Applications

that run on the REE. The TEE thus offers more security than the REE and security sensitive operations is ran in this environment. However, a TEE is not as secure as the Secure Element, if stronger security is required then secret keys can be stored on the Secure Element and sensitive operations can be carried out in this tamper resistant environment. The TEE standard defines API that allows the TEE communicate with the Secure Element called the Secure Element API [9]. Who has control over an how access is granted

As attractive as an embedded Secure Element Sounds, it is a resource restrained environment as compared to the TEE and hence has limitations for more complex resource required operations. In addition to this, TEE gives the opportunity for multiple remote authorities to have Security Domains and Trusted Applications running on a device responsible for different operations on like a Secure Element that is managed by one authority. The current state of the architecture, OEMs control Secure Element on mobile devices which then becomes a more complex business consideration for Service Providers to have access to the Secure Element.

SmartSD's could be a solution to the Secure Element control and access challenge. The SmartSD would be an external Secure Element that can be personalised for each voter with keys, biometric information specific to each voter after registration. Using the Secure Element API, the Trusted Voting Application can request for cryptographic services from the SmartSD executed within the SmartSD. Although SmartSD's are not considered in this thesis because some mobile devices such as IPhone do not have provisions for external SmartSD but it should be investigated in future works.

## 6.9 Summary

In this section we presented a novel TEE Mobile Voting Scheme that prevents ballot stuffing, voter impersonation and ineligible voting. But first an Untrustworthy Environment was defined that encompasses a Hostile Voting Environment and an Untrust-

worthy Mobile Device. The attacker's goals and possible attack vectors were presented. We then described an anonymous group signature scheme which is required in our proposed voting scheme to provide eligibility verifiability and prevent ballot stuffing.

Our proposed voting scheme runs on a mobile device with security sensitive operations such as ballot encryption and anonymous ballot signing, executed in the Trusted Execution Environment of the mobile device. The TEE provides isolated execution of security sensitive operations; secure storage of keys; secure display of ballot to voters and secure input of voter's choices by voters using the Trusted User Interface. With the trusted biometric peripherals and Biometric API of the TEE, we verify the voters identity preventing voter impersonation.

Finally we did a security analysis of our TEE Mobile Voting Scheme and show that it prevents voter impersonation, ineligible voting and ballot stuffing in our defined Untrustworthy Environment II.

# Part IV

# Untrustworthy Environment-Use Case

# Chapter 7

# Adding Eligibility Verifiability to an Existing Prêt-à-Voter Scheme

*Eligibility verifiability helps to prevent voter impersonation and ballot stuffing. End-to-End verifiable schemes allows voters check that their votes have been cast-as-intended, recorded-as-cast and observers can confirm votes have been counted-as-record. However, if voter authentication is done external to the end-to-end voting scheme in an untrustworthy environment, it creates an opportunity for voter impersonation. In this chapter we propose the addition of voter and ballot authentication using smartcards, anonymous group signature and a Trusted Execution Environment*

## 7.1 Introduction

Prêt-à-Voter is an end-to-end verifiable voting schemes that aims to be independent of the underlying technology to guarantee ballot secrecy and accurate vote count. Prêt-à-Voter schemes were designed for use in a supervised voting environment, with the poll worker authenticating the voter using traditional means before blank ballots are issued to the voter. In a standard Prêt-à-Voter ballot is split into two halves (see table 7.1),

the names of candidates is listed on the left half of the ballot in a random order. On the bottom half on the right half, is an encrypted value of the random ordering of the listed candidate on the left half of the ballot, called an onion. A voter ticks the box on the right hand side, against the name of the candidate he intends to vote for (see table 7.2), splits the ballot paper into two halves. The left half is discarded and the right half is kept by the voter as a preference receipt (see table 7.3).The election official now scans and signs the receipt as the voter's vote, using the appropriate device. The voter keeps the signed preference receipt to confirm later that his ballot as been recorded-as-cast.

Since Chaum proposed a voter verifiable scheme using visual cryptography to encyrpt receipts [41], that eventually led to Prêt-à-Voter scheme proposed in [158], there has been a lot of work done on Prêt-à-Voter schemes over the years [47, 62, 64, 152]

In this chapter we consider one of such Prêt-à-Voter schemes, "vVote: A Verifiable Voting System" [159, 62], used in the Victoria Elections in Australia. This is an End-to-end verifiable voting scheme that aims to guarantee voters that their ballots have been cast-as-intended, recorded-as-cast and counted-as-recorded. The Prêt-à-Voter ballots are printed on demand after voter authentication on election day on like in [174] where ballots are printed prior to elections, stored in envelopes and opened after voter authentication.

Nevertheless, as discussed in section 4, even if the scheme is end-to-end verifiable, because voter authentication is done external to the voting scheme it gives an opportunity for ineligible voting and ballot stuffing. We acknowledge the fact that the scheme accounts for this vulnerability and suggests that traditional means for authenticating voter should suffice considering the environment the scheme would be deployed. We argue that, if this scheme is deployed in an environment where this trust assumptions do not hold, then the scheme becomes vulnerable to electoral fraud.

In chapter 5 we proposed an electronic voting scheme based on smartcards and in chapter 6 we proposed an anonymous group signature voting scheme in the Trusted Execution Environment of a mobile device. Both schemes use biometric technology for

voter verification, to reduce the chance of voter impersonation.

In this chapter we incorporate some of the cryptographic techniques and technology used in the other schemes into the vVote scheme. We introduce a Smartcard as a representative of the voter that can authenticate the card holder and carry out cryptographic processes on behalf of the voter. We maintain the back end entities, cryptographic protocols and technology used in the vVote scheme, we only include Voter verification and Ballot authentication to the front end.

Table 7.1: Blank Prêt-à-Voter Ballot

| Candidate | Mark |
|---|---|
| John | |
| Mariam | |
| Adedare | |
| Tony | |
| Pascal | |
| | a24j7AA |

Table 7.2: Completed Prêt-à-Voter Ballot

| Candidate | Mark |
|---|---|
| John | |
| Mariam | |
| Adedare | X |
| Tony | |
| Pascal | |
| | a24j7AA |

Table 7.3: Preference Receipt Prêt-à-Voter Ballot

| Mark |
|---|
| |
| |
| X |
| |
| |
| a24j7AA |

### 7.1.1 Description of the vVote scheme

We now present a simplified version of the vVote scheme, our main concentration is on the front end part of the scheme mainly around voter authentication and ballot authentication. More detailed description of the scheme can be found here [159]

1. On elections day the voter goes to a polling station
2. The Voter, gets authenticated using insecure traditional means as discussed in section 4.3.3 and seen in figure 7.1
3. The voter is printed a ballot from the Print-On-Demand Printer
4. The candidate order is encrypted using a threshold private key, the serial number and district for each ballot is signed prior to the election by the issuing authority.
5. The printed ballot contains the printer's encryption over the candidate order and
6. The Voter audits the ballot following same audit procedure in [62] to get assurance that the random ordering of candidate was properly encrypted
7. The audited ballot is discarded and the voter is printed off a new ballot
8. The voter takes the new ballot to an electronic ballot marker (EBM) attached with a scanner
9. The scanner scans the printed ballot and the ballot is populated on the screen of the EBM
10. The voter fills the ballot using the screen of the EBM.
11. The filled ballot is sent to the Private Web Bulletin Board by the EBM
12. The Private WBB, accepts the ballot, confirms it hasn't been altered then signs it.

13. EBM prints the signed receipt, called the preference receipt. The preference receipt contains the Private WBB's over voter's preference, the district code and serial number of the ballot.

14. The Voter then compares the ordering of preference receipt with the candidate list and then discards the candidate list if both matches.

15. The voter leaves the polling unit with the preference receipt and can compare this the information printed on a Public Web Bulletin board.

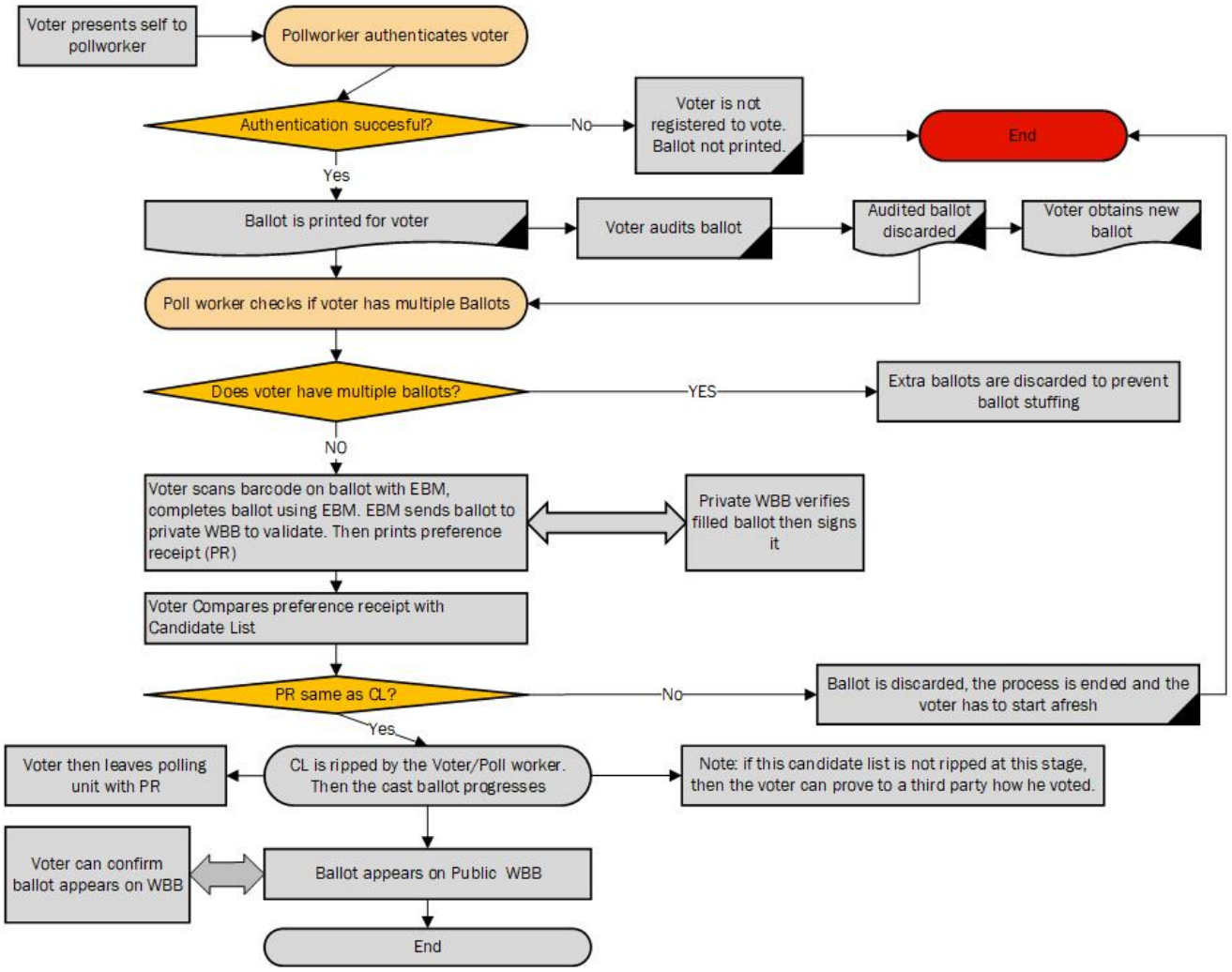


Figure 7.1: Prêt-à-Voter voting scheme

### 7.1.2 Problem Statement

In the current Prêt-à-Voter scheme as we discussed in sections 4.3.1 and seen in figure 7.1, the pollworker is meant to authenticate voters and prevent ineligible citizens from voting. The pollworker is also responsible for preventing voters from casting multiple votes. If the trust placed on poll workers is broken, then the voting system cannot tell if an ineligible citizen has voted, an abstained voter has been impersonated or ballots have been stuffed. We discussed these issues in details in section 4.3.3.

To address these issues, we need to have technical means of authenticating the voter and the ballots. Using a smartcard for authentication moves the trust placed on the poll workers to the smartcard. This is a reasonable thing to do because a smartcard is tamper resistant and can securely carry out cryptographic operations such as signature over a ballot on behalf of the voter. A smartcard can also has voter's biometric details stored on it, which it can use to authenticate the voter to ensure it's being presented by the voter it was issued to. As discussed in Sections 5.3.1, electronic ID cards issued to citizens of some countries have these capabilities, so this scheme is building on the existing infrastructure already in use.

Privacy is an important requirement of electronic voting schemes, authenticating the ballot links the voter's identity to the ballot, which could allow a third party tell how a voter voted. To address this issue an anonymous technique of signing ballots by voters needs to be adopted

### 7.1.3 Contribution

Considering the issues identified in the problem statement we adopt some of the techniques used in the protocols proposed in chapters 5 and 6 and incorporate it into a existing Prêt-à-Voter scheme.

1. We add eligibility verifiability to an existing Prêt-à-Voter scheme by introducing a smartcard as an instance of the voter to manage credentials on behalf of the

voter.

2. We use bioemtric technology to verify the card holder identity, to prevent ineligble voting by citizens or voting on behalf of an abstained voter. see section 5.4.5 and 6.6.2

3. We introduce the use of anonymous group signature described in sections 6.3.1 and 6.3.2, to tie the ballot to the voter's identity whilst still providing voter anonymity.

4. Finally we suggest the use of a Trusted Execution Environment within the purpose built Electronic Ballot Marker

## 7.2 Smartcard Prêt-à-Voter Scheme

In this section we introduce our proposed voting scheme. We use some of the technologies and cryptographic techniques discussed in chapters 5 and 6. We start by listing some of the assumptions we make about the scheme. After which we discuss key entities involved in the scheme. finally we discuss the proposed protocol and messages exchanged between entities.

### 7.2.1 Assumption of Our Voting Scheme

We now discuss some of the assumptions we make, these assumptions are reasonable because it is based on existing smartcards capabilities and GlobalPlatform TEE capability discussed in chapter 3

**A-1** We assume the smartcard is preloaded with the group public key, the group member credentials, the linking base and the smartcard generates the group member private key/member signature key. Alternatively the Issuer can generate the

group member private key/member signature key, preload it on the card and delete every copy of it so it no longer has the member signature key.

**A-2** We assume only the smartcard knows this group member signature key and it never leaves the card

**A-3** We assume the card Issuer is a Trusted Authority that communicates information to and from the Private WBB. For simplicity we assume the Private WBB as seen in the vVote Scheme [159] is managed by the Issuing Authority, hence we refer to them as the same.

**A-4** We assume the TA application in the TEE of the EBM using the Trusted User Interface (see section 3.3.2) securely displays the ballot to the voter and input by voter is also secure

**A-5** We assume the Electronic Ballot Marker is equipped with a smartcard reader or has an external smartcard reader attached to it and it has the necessary API to access the Reader

### 7.2.2 Attacker Goals

**AG-1** Vote Multiple Times: The attacker could be a legitimate voter, a poll worker or an ineligible voter. The attacker's goal is to cast multiple ballots without being detected.

**AG-2** Voter Impersonation: The goal of the attacker is to cast ballots on behalf of abstained voters.

**AG-3** Ineligible Voting: The goal of the attacker is to cast ballots in elections he is not registered to vote in.

### 7.2.3 Protocol Goals

**G-1** Eligibility Verifiability: the goal of the protocol is to prevent impersonation of eligible voters and ineligible citizens from casting a ballot.

**G-2** Prevent Ballot Stuffing: The protocol intends to prevent eligible voters from casting multiple votes and corrupt poll workers from ballot stuffing

### 7.2.4 Key Entities

We now introduce some of the key entities of our Voting scheme

1. Printer: This device prints the Prêt-à-Voter ballot on demand using same techniques as described in [159, 62] and offering the same level of security.

2. Voters Card (VC): This is a smartcard and a representative of the voter, it stores secret keys, can generate random numbers and carry out cryptographic operations on behalf of the voter. Every Voter's card has a Unique group member private/signature key. The smartcard also securely stores Voter's biometric details and other personal voter information such as the district voter is eligible to vote in.

3. Issuing Authority (IA): This authority issues the voter with a Voter's Card. The Issuing Authority generates keys and stores on the card. The Issuing authority is also the Group Manager of an anonymous group signature scheme and issues a group member signature key to the Voter's smartcard (see section 6.3.1) or in an exchange of some parameters the smartcard generates the signature key. The Private WBB part of the issuer, is a secure database that receives messages, performs basic validity checks, and returns a signature. Validly signed messages later appear on the Public WBB.

4. Electronic Ballot Marker (EBM): We assume this device is equipped with sensors to capture biometric details and card reader that can read the voter's smartcard. This device is also TEE compliant as described in section 6.6.2, and as such does remote attestation to give integrity guarantees about the operating system and voting application running on the device. The TEE also offers a Trusted User Interface [83] that provides secure displays and secure inputs to the voter.

### 7.2.5 On Election Day

We now breakdown our voting protocol into the various steps.

#### Voter Registration

Prior to the day of the election, the voter goes to a registration centre for registration. The voter's eligibility is confirmed, the district the voter is eligible to vote is confirmed, then the Voter is issued a Voter's Card. After which, the voter can then go to a polling unit come election day to cast a ballot.

#### Election Day

The voter proceeds to any polling unit of his choice to exercise his franchise following the voting procedure highlighted in figure 7.2 and 7.3, which we explain below:

1. Following similar procedures in [159] the voter is printed a ballot on demand, the voter can decide to audit this ballot to ensure the random candidate ordering of the candidate list has been encrypted correctly.

2. If the voter audits the ballot, that ballot is discarded and a fresh ballot is printed off for the voter

3. The Voter then obtains a fresh ballot, the ballot is captured with the electronic ballot marker using the scanner attached to it to scan the barcode on the ballot in **step 1** and **step 2** of the protocol diagram.

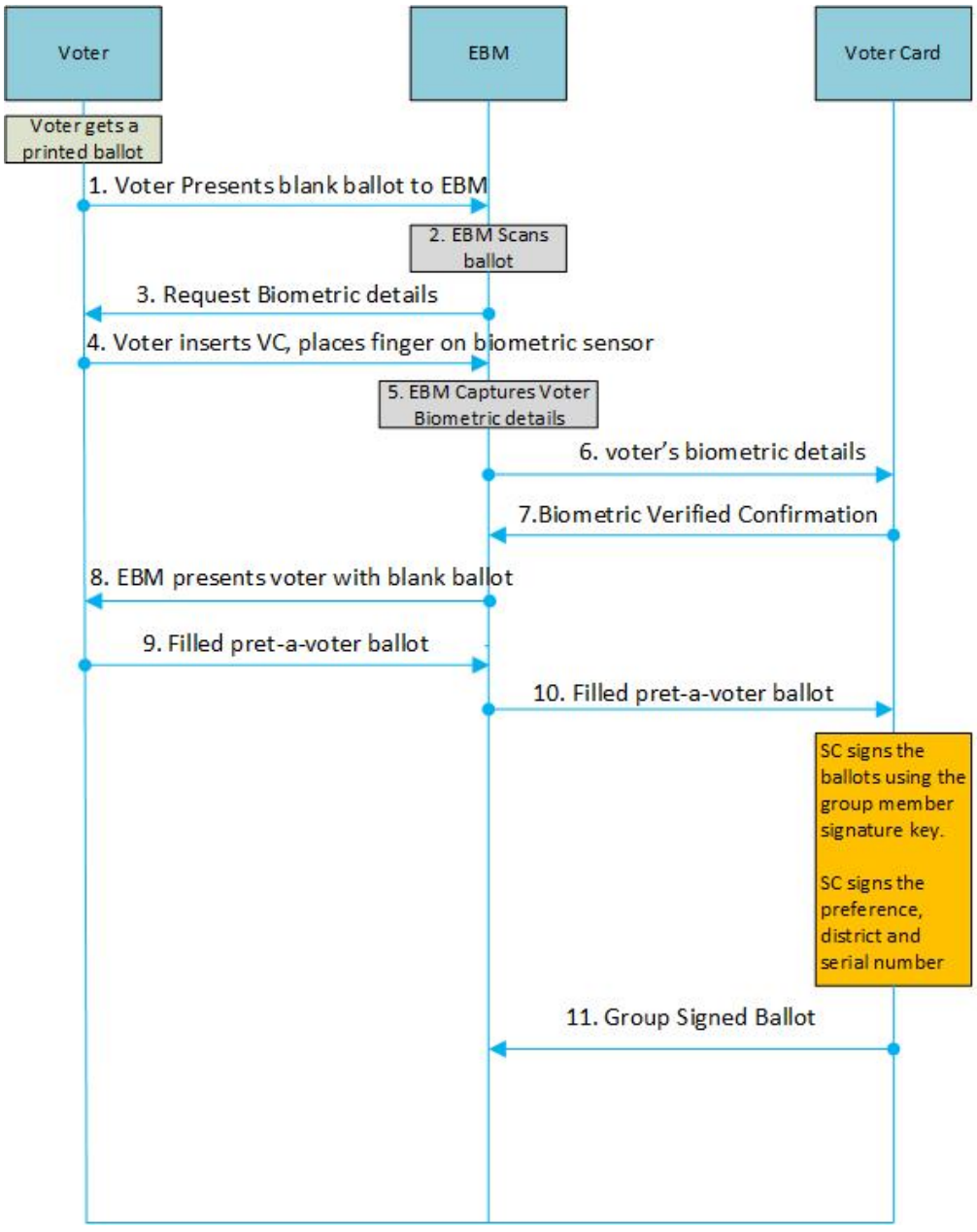


Figure 7.2: Smartcard Prêt-à-Voter Voting Protocol

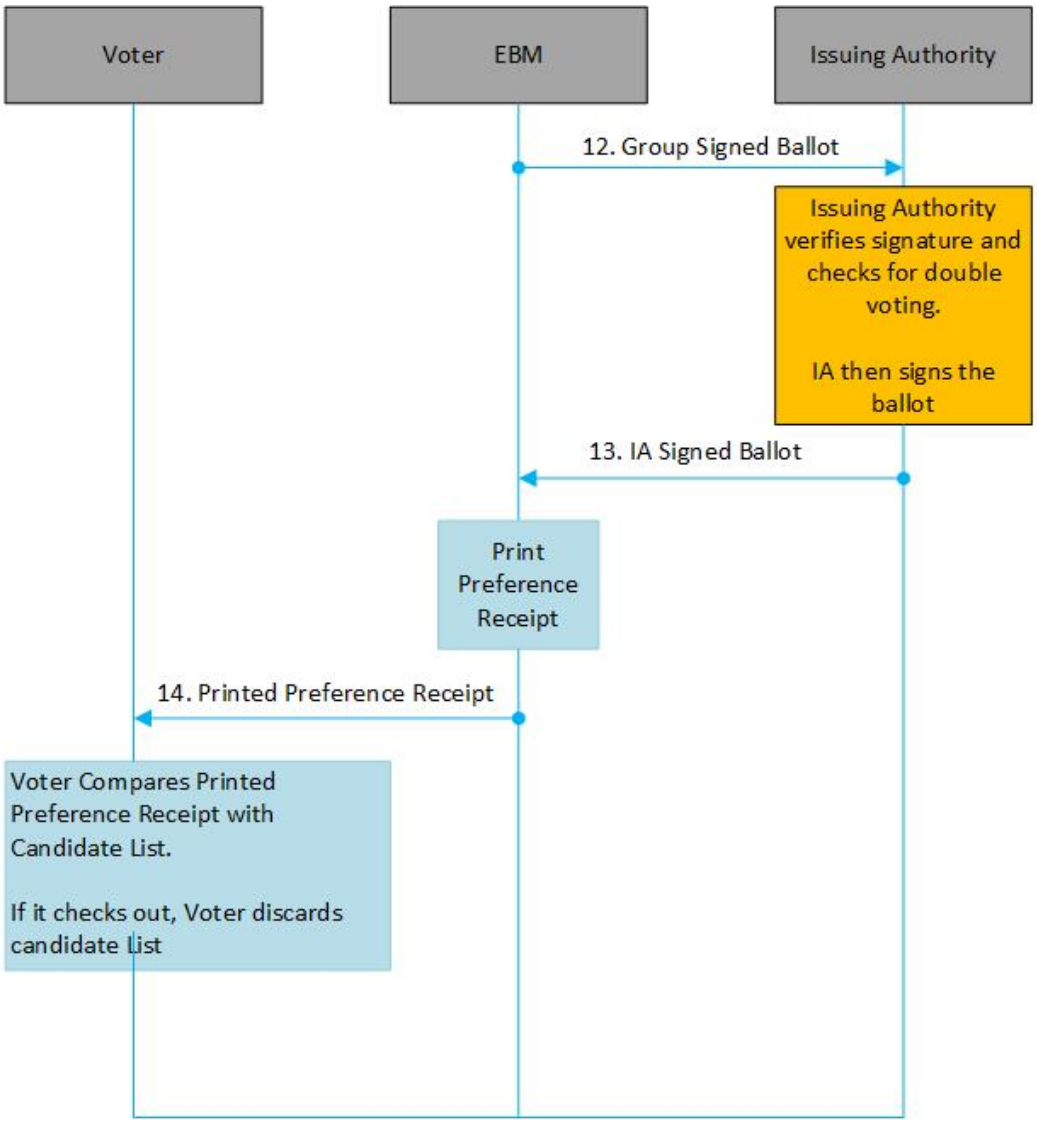


Figure 7.3: Smartcard Prêt-à-Voter Voting Protocol (continued)

4. We assume this ballot has card reader and the ability to capture a voter biometric data. The EBM requests for the voter's biometric data in **step 3**.

5. The voter inserts his Voter's card in **step 4** and then presents his biometric details i.e. places fingers on biometric scanner.

6. In **step 5**, the EBM captures the voter biometric details and securely sends this details to the Voter's card for confirmation in **step 6** using trusted peripherals as specified in the GPD Standard [83] and discussed in chapter 4

7. The voter biometric details is verified by the card, by matching it on-card as described in section 5.4.2 and 5.4.5, if this is incorrect, the processes is aborted, if not a biometric verified confirmation is sent back to the EBM in **step 7**.

8. EBM then populates the scanned ballot and securely displays it to the voter on the screen in **step 8** using the Trusted User Interface of the TEE.

   In our assumptions, We assumed this device has a Trusted Execution Environment running in parallel to the untrustworthy Rich Execution Environment on the EBM.

9. The voter fills out the ballot in **step 9**, the TEE TUI's Secure Input capability (see section 3.3.2 for TUI). The EBM sends this ballot to VC in **step 10**

10. On receipt of the filled ballot, VC can confirm the district number on the ballot is same as the voter is registered to and stored on the card. If it isn't, VC sends a rejection message to EBM, EBM discards the ballot and the process is aborted. If the district matches, VC signs an anonymous group signature over the serial number, Voter's district and Voter's preference we call the group signed ballot asper figure 7.2. VC then sends the group signed ballot to EBM in **step 11**.

11. The group signed ballot is forwarded to the Issuing Authority by EBM in **step 12**

12. The Issuer then verifies the validity of the anonymous group signature over the ballot to confirm the voter is an eligible voter registered to a group without knowing the voter's identity. If the signature verification fails, the ballot is discarded and the process is aborted.

    With the linking base, IA verifies if the voter has voted multiple times and discards duplicate ballots if the voter has voted multiple times to prevent ballot stuffing.

13. The Issuing Authority then signs over the ballot and sends this back to the EBM in **step 13**. IA posts this information on web bulletin board. IA specifically signs over the Voter's preference, District, Ballot serial number, Group signed ballot by the Voter

14. The EBM prints out the preference receipt with the Issuing Authority's signature over the receipt and the voter retrieves the preference receipt in **step 14**

15. The voter compares the preference receipt and candidate list, if both matches, voter is assured ballot has been cast successfully

16. The ballot on the Private WBB then goes through a re-encryption mix-net for shuffling and eventual decryption following similar procedure as [158, 159]

## 7.3 Security Analysis

We now do a security analysis to show how the security goals have been met

1. **G-1** Eligibility Verifiability: After the EBM scans the ballot, the EBM requests for voter's biometric details in step 3 of the protocol. The EBM captures the

voter's biometric details and sends to VC. This whole process is done using the Trusted User Interface and Biometric API of the EBM provided by the GPD Standard [83] and discussed in section 3.3.2. VC matches the captured biometric detail with the template stored on the card, verifying the Voter's Identity. The voter biometric template stored on the card, never leaves the tamper resistant card.

We assume the biometric sensor of the EBM has a liveness checking mechanism, for example the EBM sensor has to feel a finger in the case of finger print biometrics before Voter's Biometric detail is captured. For an attacker to cast votes on behalf of an abstained voter ( **AG-2**), even if the attacker is in possession of the Voter's, Card the attacker and presents his biometric details, when VC tries to match this with the details stored on the card it fails and the ballot is rejected, still satisfying our security goal goal1.

Furthermore, even if the poll worker is potentially corrupt, authenticates an unregistered citizen and hands him a ballot paper ( **AG-3**), the unregistered citizen still cannot successfully cast a ballot without having a legitimate Voter's card and the accompanying bioemtric details.

2. **G-2** Prevent Ballot Stuffing: In step 2, the EBM scans and captures the ballot in step 2, after the voter has been authenticated in step 7, EBM populates the ballot on the screen for the voter to complete. In step 10, EBM sends the completed ballot to VC, VC anonymously signs the ballot using his group signature key and sends it to the Issuing Authority via the EBM. Issuing Authority passes the ballot via the group signature verification algorithm using the group public key to verify it's validity. If the signature verification fails the ballot is discarded and process aborted. If VC's signature is valid, using the linking base attached to the signature VC can verify if a voter has voted twice without knowing the voter's identity. If the voter has voted twice, the duplicate ballot is discarded

hence prevent ballot stuffing which satisfies our protocol goal **G-2**.

As stated earlier, the district the voter is registered to is stored on the card and appears on the ballot. If a legitimate voter attempts to vote in a wrong election, when VC receives the completed ballot, if the district on the ballot doesn't match that stored on VC, VC rejects the ballot and the process is aborted. This helps to prevent voter's that have been properly authenticated from casting ballot in districts they haven't been registered to, further preventing ballot stuffing.

## 7.4 Discussion

In the vVote scheme, election officials are expected to enforce the destruction of the candidate list after voter's have confirmed that the preference receipt matches the candidate list. If voters are allowed to leave the polling unit with the candidate list then they can prove to a third party how they voted. In this scheme we do not offer a technical solution to address this issue, we maintain same assumption for preventing this as the original vVote scheme. We mainly focus on moving the trust from poll workers authenticating voters to a smartcard (Voter's Card). With the stored biometric details of the voter on the Voter's card, VC can can authenticate the voter

Furthermore, the poll workers are trusted to prevent ballot stuffing, by ensuring voters are only given one ballot. We have moved that trust from the poll worker to the voter's Card and Issuing Authority. The Voter's Card anonymously signs a completed ballot on behalf of the voter and sends this to the Issuer via the EBM. Issuing Authority verifies the anonymous signature and checks for multiple ballots without knowing the identity of the signer. Issuing Authority can then delete duplicate ballots thereby preventing ballot stuffing.

For the rest of the scheme we do not alter any of the procedures in the vVote scheme [159, 62] used for ballot generation, print-on-demand procedures, shuffling ballots, displaying information on the Public Web Bulletin Board, ballot audit and ballot

cancellation procedures etc. We also do not change procedures for generating election keys and random numbers. Threshold assumptions and all other cryptographic assumptions made in the vVote scheme still holds. We simply just include technical means of authenticating voters, authenticating ballot and linking that to the existing procedure of the vVote scheme.

The ballots used in the current vVote scheme is a complicated complicated Prêt-à-Voter ballot that contains many candidates and EBM was meant to make filling out the ballots easier. Our scheme introduced is quite generic hence we do not consider the exact content of the ballot, but we assume the scheme can accommodate Prêt-à-Voter ballots and the complexity of the ballot would be election specific.

We acknowledge the fact that vVote scheme was designed for a different environment and some of the threats we have identified may not necessarily be an issue in that environment. As an example, every voter is mandated by law to vote in elections in Australia where the vVote scheme was designed for. That would imply that there would be no abstained voters, hence no one would impersonate an abstained voter. However if this same end-to-end verifiable scheme is deployed in an environment where trust for human procedural voter authentication does not hold, then the scheme becomes vulnerable to electoral. Our scheme aims to address this issue.

vVote scheme was also designed for use by visually impaired voters, in our proposed scheme we do not consider how visually impaired voters can interact with it. This could be looked into in future works.

## 7.5 Summary

Election is at the heart of a countries democracy, if the citizens do not trust the electoral process it could lead to voter apathy. End-to-End electronic voting scheme gives voters assurance that their votes have been cast-as-intended, recorded-as-cast and counted-as-recorded. However, if voter authentication is external to the end-to-end verifiable

voting scheme then it introduces avenues for voter impersonation and ballot stuffing.

In this chapter we have proposed a voting scheme that adds internal authentication to an existing end-to-end verifiable scheme. In our scheme, voters are issued with a Voter's Card that can authenticate the cardholder using biometric technology. The card can also sign ballots whilst preserving the identity of the voter by using an anonymous group signature scheme defined in section 6.3.1.

Finally we did a security analysis to show that our scheme prevents voter impersonation and ballot stuffing, satisfying the protocol goals defined in section 7.2.3.

# Chapter 8

# Conclusion and Future Work

This thesis sets out to propose electronic voting schemes for use in untrustworthy voting environments with an aim to mitigate electoral fraud such as ballot stuffing, ineligible voting and voter impersonation that have plagued real world elections in these environments.

The thesis starts of my presenting a literature of electronic voting schemes that have been proposed over the years and the cryptographic mechanisms needed to protect the integrity of the elections. In chapter 2, we also introduced anonymous group signature schemes, a cryptographic mechanism that allows members of a group anonymously sign messages that can be verified by a group manager whilst their identity remains anonymous. Anonymous group signature is a pivotal part of the protocols proposed in chapters 6 and 7.

The secure platform problem makes it difficult to propose electronic voting schemes on a mobile device because the outcome cannot be guaranteed due to malwares that may be in control of the mobile device. This is why we define mobile devices as an untrustworthy environment in chapter 6. Ronald Rivest created the phrase "Secure Platform Problem" [82] to highlight the difficulty in protecting an insecure mobile device from malwares and corresponding attacks they may perpetuate. In chapter 3 a technical background on Global Platform Trusted Execution Environment is presented

to address the Secure Platform Problem. The Trusted Execution Environment is a hardware isolated area within a mobile device where security sensitive operations of a Client Application is executed. The scheme proposed in chapter 6 is based on the capability of the TEE to provide security in an untrustworthy mobile device even when infected with malwares.

In this thesis we considered reports of electoral fraud in some real world elections to define two untrusworthy environments. **Untrustworthy Environment I** is an untrusworthy supervised voting environment defined in chapters 4 and 5, in this environment reliance on the trustworthiness of poll workers is not suitable considering reported electoral fraud. With this in mind, a threat model was built in chapter 4 and two electronic voting schemes was analysed. After analysis, it was highlighted that both voters and poll workers could be incentivised with financial gain to cheat the system due to socio-economic issues that exists in these real world environments. It was then concluded that technical security should be leveraged on to provide voter and ballot authentication to prevent ballot stuffing, voter impersonation, ineligible voting and vote selling in this environment.

In chapter 5, **Untrustworthy Environment I** was formally defined, threats in this environment highlighted and capabilities of the attacker defined. Based on the threat model a generic mutual authentication electronic voting scheme was proposed. The proposed scheme uses smartcards as an instance of the voter to carry out cryptographic operations on behalf of the voter. A biometric template is stored on the card for voter verification. Using digital signatures and nonces, the smartcard mutual authenticates the Voting Terminals and card Issuing Authority. The scheme is incorporated into a reencryption mixnet scheme and a security analysis shows the scheme provides eligibility verifiability and satisfies other security requirements of an electronic voting scheme in **Untrustworthy Environment I**. A mechanical formal analysis using Scyther shows that authentication which underpins eligibility verifiability is not broken in our generic mutual authentication scheme, hence ballot stuffing and voter imperson-

ation is prevented which is the main goal of the protocol. Satisfying the protocol goals and addresses research question **RQ-2**

As earlier discussed, the secure platform problem makes it difficult to guarantee the outcome elections in mobile devices. This makes the mobile device an untrustworthy environment. We defined a Hostile Voting Environment based on various reports about electoral violence from real world elections that has marred the integrity of elections and in some cases led to loss of lives. A Combination of the Insecure Mobile Device and Hostile Voting Environment leads us to define an **Untrustworthy Environment II**. We made a case for voting using a mobile device in this hostile environment since billions of smart mobile devices are used worldwide. However, we still have to deal with the secure platform problem, to address this some voting schemes proposed in the literature have suggested code voting [50, 107]. In chapter 6, we propose a TEE mobile voting scheme that attempts to address the Secure Platform Problem and research question **RQ-3**.

Our TEE Mobile Voting Scheme in chapter 6, leverages on the security capabilities of the TEE architecture to provide security within an insecure mobile device. In particular, the Trusted User Interface provides secure display of ballots and secure input through which voter's input their choices. Security sensitive operations such as ballot encryption and signing ballots is executed in TEE, isolated from insecure Rich Execution Environment. The trusted biometric peripherals and biometric API are used used to securely verify the voter's identity preventing voter impersonation. The integrity of the Trusted OS and Trusted Voting Application within the TEE is measured and verified through a Chain of Trust starting from a hardware enabled Root of Trust. Anonymous group signing keys are generated and stored securely within the TEE using the TEE Secure Storage. The Trusted Voting Application uses this group signature key to anonymously sign ballots on behalf of the voter. With the linking capability of the group signature scheme, the issuing authority can prevent ballot stuffing; confirm voter's eligibility whilst the voter's identity remains anonymous. A security analysis of

this scheme is carried out to show that it prevents ballot stuffing, voter impersonation and ineligible voting, hence satisfying our research question **RQ-3** and **RQ-4**.

End-to-end verifiable schemes as earlier defined allows voters verify their ballots have been cast-as-intended, recorded-as-count and any interested party can confirm ballots have been counted-as-recorded. In chapter 4, we did an analysis on a Prêt-à-Voter scheme used in elections in Australia called vVote, in our already defined Untrustworthy Environment. We showed from our analysis that to deploy this scheme in an untrustworthy environment trust needs to be moved from poll workers to a more secure device. We also concluded that authentication should be internal to the protocol and not external. In chapter 7, we proposed a scheme that adds eligibility verifiability to the current vVote infrastructure addressing research question **RQ-2**. This is achieved by issuing smartcards to voters, that can anonymously sign ballots and verify voter's identity using biometric template stored on the smartcard. A security analysis is done to show that our proposed smartcard Prêt-à-Voter scheme, prevents ballot stuffing,voter impersonation and ineligible voting whilst the voter's identity remains anonymous in an Untrustworthy Environment satisfying research question **RQ-2**

## 8.1 Future Works

In this thesis we proposed three electronic voting schemes for different voting environment. We present below areas in these schemes that could benefit from further works.

1. Formal Verification of proposed schemes: In our mutual authentication voting scheme proposed in chapter 5, a mechanical formal protocol analysis using Scyther was done to show the protocol was formed correctly and mutual authentication between the various entities was not broken, hence eligibility verifiability was satisfied. In future works, a formal verification should be done on the protocol when it is incorporated into a re-encryption mixnet to prove that security

properties like receipt freeness, privacy and verifiability is achieved.

In chapter 7, we introduced a smartcard as an instance of the voter that can anonymously sign ballots and verify voter's identity using the biometric template stored on the card into an existing Prêt-à-Voter. The existing scheme trusts poll workers to authenticate voter, in our proposed scheme that trust was moved to a tamper resistant smartcard, making authentication internal to the Prêt-à-Voter scheme [159]. However, our Smartcard Prêt-à-Voter has not been formally proven, this is an area for future work.

2. Implementation of Proposed Schemes: In chapter 6, the TEE Mobile Voting Scheme was proposed and security analysis done to show it satisfies eligibility verifiability and other electronic Voting security requirements. However, implementation of the scheme was not done to show the functionality of the TEE. In future works, the TEE Mobile Voting scheme should be implemented to show the interactions between the Voting Application in the REE and Trusted Applications in the TEE. The implementation should also show the functionality of the Trusted User Interface in providing secure display and secure input. An analysis of the implemented scheme should be done to check it's usability, performance, functionality and security.

   In chapter 7, a smartcard was introduced to an existing Prêt-à-Voter scheme (vVote),to add eligibility verifiability to the existing infrastructure. This new addition should be implemented in future works to investigate the functionality, performance, security and usability of Smartcard Prêt-à-Voter scheme.

3. Consumer Centric TEE: The TEE Mobile Voting Scheme proposed in chapter 6, security sensitive operations are executed in the TEE to offer more security than the Rich Execution Environment. A SmartSD is a Secure Element that offers the same level of security as a tamper-resistant smartcard. A SmartSD can be

personalized for each voter, to contain secret information unique to the voter. The possibility of using a SmartSDs in current mobile devices to run the TEE Mobile Voting Scheme and its implementation should be explored further in future works.

4. With the growth of cloud computing and it's resource capability, voting schemes should leverage on this to reduce cost of purchasing and maintaining voting infrastructures. However, this comes with a different set of security risks considering the sensitivity of binding elections. How voting schemes can be integrated into cloud technology, to take advantage of the resource capability provided by the cloud whilst still achieving the security requirements of an electronic voting scheme, should be explored in future works.

# Bibliography


[1] Smart Cards and Biometrics in privacy-sensitive. Smart Card Alliance, May 2002. 124

[2] Globalplatform technology, tee client api specification 1.0. version 1.0, jul 2010. 52

[3] Information technology -security techniques- anonymous digital signature. standard bs/iso 20008-1:2013, December 2013. 11, 46, 134, 135, 136, 137

[4] Information technology -security techniques- anonymous digital signature. standard bs/iso 20008-2:2013, November 2013. 46, 134, 135, 136, 146

[5] Globalplatform device technology, tee management framework. version 1.0., 2016. November. 51, 52, 54, 63, 64, 144, 149

[6] Globalplatform device tee initial configuration version 1.1, November 2016. 67

[7] Globalplatform device technology, tmf: Asymmetric cryptography security layer. version 1.0, 2017. March. 51, 152, 154, 155

[8] Globalplatform technology, root of trust definitions and requirements version 1.1, jun 2018. 66

[9] Globalplatform technology tee internal core api specification version 1.1.2.50, June 2018. 59, 60, 169

[10] Globalplatform technology, tee trusted user interface low-level api. version 1.0, mar 2018. 51

[11] Gheith A. Abandah, Khalid A. Darabkh, Tawfiq Ammari, and Omar Qunsul. Secure National Electronic Voting System. *J. Inf. Sci. Eng.*, 30(5):1339–1364, 2014. 33, 70, 86, 95, 98

[12] Masayuki Abe. Universally verifiable mix-net with verification work independent of the number of mix-servers. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 437–447, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. 37, 40

[13] A. Abhyankar and S. Schuckers. Fingerprint Liveness Detection Using Local Ridge Frequencies and Multiresolution Texture Analysis Techniques. In *2006 International Conference on Image Processing*, pages 321–324, Oct 2006. 92

[14] Alessandro Acquisti. An user-centric mix-net protocol to protect privacy. In *In Proc. of the Workshop on Privacy in Digital Environments: Empowering Users*, 2002. 38

[15] Ben Adida. Helios: Web-based Open-Audit Voting. In *Proceedings of the 17th USENIX Security Symposium*, pages 335–348, San Jose, CA, USA, July 2008. 49, 122

[16] Ben Adida, Olivier De Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a university president using open-audit voting: Analysis of real-world use of helios. In *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'09, pages 10–10, Berkeley, CA, USA, 2009. USENIX Association. 49

[17] Ben Adida and Ronald L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *Proceedings of the 2006 ACM Workshop on Privacy in*

*the Electronic Society, WPES 2006, Alexandria, VA, USA, October 30, 2006*, pages 29–40, 2006. 33, 34

[18] Riza Aditya, Colin Boyd, Ed Dawson, and Kapali Viswanathan. Secure e-voting for preferential elections. In *Electronic Government, Second International Conference, EGOV 2003, Prague, Czech Republic, September 1-5, 2003, Proceedings*, pages 246–249, 2003. 43

[19] Riza Aditya, Byoungcheon Lee, Colin Boyd, and Ed Dawson. Implementation issues in secure e-voting schemes. 2004. 30

[20] R. Anane, R. Freeland, and G. Theodoropoulos. e-voting requirements and implementation. In *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*, pages 382–392, July 2007. 31, 32

[21] Apple. Ios seccurity, ios 12.1. Technical report, November 2018. 138, 163, 168

[22] Ghada Arfaoui, Said Gharout, and Jacques Traoré. Trusted execution environments: A look under the hood. In *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2014, Oxford, United Kingdom, April 8-11, 2014*, pages 259–266, 2014. 57

[23] ARM. Arm security technology building a secure system using trustzone® technology. techreport, December 2009. 51, 67

[24] Joseph Asunka, Sarah Brierley, Miriam Golden, Eric Kramon, and George Ofosu. Protecting the polls : The effect of observers on election fraud 1. 2013. 71, 74, 82

[25] Indepent National Electoral Authority. *Regulations and Guidelines for The Conduct of Elections*. Indepent National Electoral Authority, Nigeria, January 2019. 28, 131

[26] Fabrizio Baiardi, Alessandro Falleni, Riccardo Granchi, Fabio Martinelli, Marinella Petrocchi, and Anna Vaccarelli. SEAS, A Secure e-voting Protocol: Design and Implementation. *Computers & Security*, 24(8):642–652, 2005. 97

[27] Harald Baldersheim and Norbert. Kersting. *Electronic voting and democracy : a comparative analysis / edited by Norbert Kersting and Harald Baldersheim.* Palgrave Macmillan New York, 2004. 28

[28] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. Nist special publication 800-57. *NIST Special publication*, 800(57):1–142, 2007. 107

[29] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*, PODC '01, pages 274–283, New York, NY, USA, 2001. ACM. 161

[30] Jonathan Ben-Nun, Niko Fahri, Morgan Llewellyn, Ben Riva, Alon Rosen, Amnon Ta-Shma, and Douglas Wikström. A New Implementation of a Dual (Paper and Cryptographic) Voting System. In *5th International Conference on Electronic Voting EVOTE*, pages 315–329, Castle Hofen, Bregenz, Austria, July 2012. 97

[31] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 544–553, New York, NY, USA, 1994. ACM. 28, 31, 40, 41, 42

[32] Michael Bratton. Vote buying and violence in nigerian election campaigns. *Electoral Studies*, 27(4):621 – 632, 2008. 19, 130, 131

[33] Mike Burmester and Emmanouil Magkos. Towards secure and practical e-elections in the new era. In *Secure Electronic Voting*, pages 63–76. 2003. 32

[34] Craig Burton, Chris Culnane, and Steve Schneider. Secure and verifiable electronic voting in practice: the use of vvote in the victorian state election. 04 2015. 80

[35] Published by S. O'Dea and Sep 13. Cell phone sales worldwide, Sep 2021. 21, 138, 139

[36] Sébastien Canard, Berry Schoenmakers, Martijn Stam, and Jacques Traoré. List signature schemes. *Discrete Applied Mathematics*, 154(2):189 – 201, 2006. Coding and Cryptography. 45, 46, 134

[37] Richard Carback, David Chaum, Jeremy Clark, John Conway, Aleksander Essex, Paul S. Herrnson, Travis Mayberry, Stefan Popoveniuc, Ronald L. Rivest, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. Scantegrity II municipal election at takoma park: The first E2E binding governmental election with ballot privacy. In *19th USENIX Security Symposium, Washington, DC, USA, August 11-13, 2010, Proceedings*, pages 291–306, 2010. 49

[38] Thomas E. Carroll and Daniel Grosu. A secure and anonymous voter-controlled election scheme. *J. Network and Computer Applications*, 32(3):599–606, 2009. 34, 38, 40, 43

[39] O. Cetinkaya. Analysis of security requirements for cryptographic voting protocols (extended abstract). In *2008 Third International Conference on Availability, Reliability and Security*, pages 1451–1456, March 2008. 31, 32

[40] Ashish Chaturvedi. Rigging elections with violence. *Public Choice*, 125(1/2):189–202, 2005. 19, 130

[41] D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security Privacy*, 2(1):38–47, Jan 2004. 33, 40, 173

[42] D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security Privacy*, 2(1):38–47, Jan 2004. 34

[43] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981. 28, 37, 38, 39

[44] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, pages 199–203, Boston, MA, 1983. Springer US. 43

[45] David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, and Alan T. Sherman. Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *2008 USENIX/ACCURATE Electronic Voting Workshop, EVT 2008, July 28-29, 2008, San Jose, CA, USA, Proceedings*, 2008. 34

[46] David Chaum, Bert den Boer, Eugène van Heyst, Stig Fr. Mjølsnes, and Adri Steenbeek. Efficient offline electronic checks (extended abstract). In *Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*, pages 294–301, 1989. 45

[47] David Chaum, Peter Y. A. Ryan, and Steve Schneider. A practical voter-verifiable election scheme. In Sabrina de Capitani di Vimercati, Paul Syverson, and Dieter Gollmann, editors, *Computer Security – ESORICS 2005*, pages 118–139, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 33, 40, 173

[48] Mohammad Jabed Morshed Chowdhury. Article: Comparison of e-voting schemes: Estonian and norwegian solutions. *International Journal of Applied Information Systems*, 6(2):60–66, September 2013. Published by Foundation of Computer Science, New York, USA. 28, 33

[49] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. In *IEEE Symposium on Security and Privacy (S&P 2008)*, pages 354–368, Oakland, California, USA, May 2008. 28, 97

[50] Sheila Cobourne, Lazaros Kyrillidis, Keith Mayes, and Konstantinos Markantonakis. Remote e-voting using the smart card web server. *IJSSE*, 5(1):39–60, 2014. 193

[51] Sharon B Cohen. Auditing technology for electronic voting machines. 06 2006. 29

[52] Eric Cole. Insider threat in law enforcement. sans white paper. sans institute, 2014. 75

[53] Susan Collard. The Expatriate Vote in the French Presidential and Legislative Elections of 2012: A Case of Unintended Consequences. *Parliamentary Affairs*, 66(1):213–233, 01 2013. 48

[54] Paul Collier and Pedro C. Vicente. Votes and violence: Evidence from a field experiment in nigeria. *The Centre for the Study of African Economies Working Paper Series*, 124, 01 2008. 19, 130

[55] Véronique Cortier, Georg Fuchsbauer, and David Galindo. Beleniosrf: A strongly receipt-free electronic voting scheme. *IACR Cryptol. ePrint Arch.*, 2015:629, 2015. 35, 36

[56] Véronique Cortier, Pierrick Gaudry, and Stéphane Glondu. Belenios: a simple private and verifiable electronic voting system. In *Foundations of Security, Protocols, and Equational Reasoning*, pages 214–238. Springer, 2019. 35, 36

[57] Veronique Cortier and Cyrille Wiedling. A formal analysis of the norwegian e-voting protocol. *Journal of Computer Security*, 25(1):21–57, 2017. 49

[58] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. *European Transactions on Telecommunications*, 8(5):481–490, 1997. 41

[59] L. F. Cranor and R. K. Cytron. Sensus: a security-conscious electronic polling system for the internet. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, volume 3, pages 561–570, Jan 1997. 28, 43, 44, 97

[60] Cas Cremers and Sjouke Mauw. Operational Semantics of Security Protocols. In Stefan Leue and Tarja Johanna Systä, editors, *Scenarios: Models, Transformations and Tools*, pages 66–89, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 116

[61] Cas J. F. Cremers. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In Aarti Gupta and Sharad Malik, editors, *Computer Aided Verification*, pages 414–418, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 116

[62] Chris Culnane, Peter Y. A. Ryan, Steve A. Schneider, and Vanessa Teague. vVote: A Verifiable Voting System. *ACM Trans. Inf. Syst. Secur.*, 18(1):3:1–3:30, 2015. 3, 35, 71, 75, 92, 97, 120, 122, 173, 175, 181, 188

[63] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying properties of electronic voting protocols. In *in "Proceedings of the IAVoSS Workshop On Trustworthy Elections (WOTE'06*, pages 45–52, 2006. 31

[64] Denise Demirel, Maria Henning, Jeroen van de Graaf, Peter Y. A. Ryan, and Johannes Buchmann. Prêt à voter providing everlasting privacy. In James Heather, Steve Schneider, and Vanessa Teague, editors, *E-Voting and Identify*, pages 156–175, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 173

[65] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 609–626, 2004. 46

[66] D. Dolev and A. C. Yao. On the security of public key protocols. In *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, SFCS '81, pages 350–357, Washington, DC, USA, 1981. IEEE Computer Society. 116

[67] Jorge I. Domínguez and James A. McCann. Mexicans react to electoral fraud and political corruption: An assessment of public opinion and voting behavior. *Elsevier*, Jan 1, 1998 1998. 35, 70, 74

[68] Jerome Dossogne and Frederic Lafitte. Blinded additively homomorphic encryption schemes for self-tallying voting. *Journal of Information Security and Applications*, 22:40 – 53, 2015. Special Issue on Security of Information and Networks. 161

[69] Lovi Dua and Divya Bansal. Review on mobile threats and detection techniques. *International Journal of Distributed and Parallel systems*, 5:21–29, 07 2014. 133

[70] Morris Dworkin. Nist sp 800-38d recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac, November 2007. 155

[71] Anthony Egobueze and Callistus Ojirika. Electoral violence in nigeria's fourth republic: Implications for political stability. *Journal of Scientific Research and Reports*, 13:1–11, 01 2017. 19, 130

[72] Marisa Ensor. Paul collier. wars, guns, and votes: Democracy in dangerous places , new york, ny: Harper collins, 2009. *African Conflict and Peacebuilding Review*, 1:161–164, 04 2011. 19, 130

[73] Chun-I Fan and Wei-Zhe Sun. An efficient multi-receipt mechanism for uncoercible anonymous electronic voting. *Mathematical and Computer Modelling*, 48(9):1611 – 1627, 2008. Mathematical Modeling of Voting Systems and Elections: Theory and Applications. 45

[74] Abiodun Fatai and Lekan Adisa. The use of biometric technology in the success of the 2015 general elections in nigeria. *Politeia*, 36, 10 2017. 20

[75] Jessica A Fay. Elderly electors go postal: Ensuring absentee ballot integrity for older voters. *Elder LJ*, 13:453, 2005. 29

[76] Yang Feng, Siaw-Lynn Ng, and Scarlet Schwiderski-Grosche. An electronic voting system using gsm mobile technology. 2006. 45

[77] Karen E. Ferree, Clark C. Gibson, and James D. Long. Voting behavior and electoral irregularities in kenya's 2013 election. *Journal of Eastern African Studies*, 8(1):153–172, 2014. 74

[78] Springall D. Finkenauer, T. Durumeric, Z. Kitcat, J. Hursti, H. MacAlpine, M., and J.A. Halderman. Security analysis of the estonian internet voting system. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 703–715. ACM, New York, NY, USA, 2014. 19, 20, 33, 70, 71, 75, 77, 80, 83, 84, 87, 106

[79] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques*, pages 244–251, Gold Coast, Queensland, Australia, December 1992. 28, 31, 32, 43, 97

[80] Alan Gelb and Anna Diofasi. Biometric elections in poor countries: Wasteful or a worthwhile investment?: Biometric elections in poor countries. *Review of Policy Research*, 02 2019. 20, 48, 124, 125, 131

[81] Alan Gelb and Anna Diofasi Metz. *Identification revolution: Can digital ID be harnessed for development?* Brookings Institution Press, 2018. 124

[82] Ed Gerck, C. Andrew Neff, Ronald L. Rivest, Aviel D. Rubin, and Moti Yung. The business of electronic voting. In *Financial Cryptography, 5th International Conference, FC 2001, Grand Cayman, British West Indies, February 19-22, 2002, Proceedings*, pages 234–259, 2001. 129, 132, 191

[83] GlobalPlatform. Globalplatform technology, tee system architecture version 1.2, November 2018. 11, 51, 52, 54, 56, 57, 58, 59, 60, 62, 144, 145, 159, 160, 168, 182, 185, 187

[84] Philippe Golle, Sheng Zhong, Dan Boneh, Markus Jakobsson, and Ari Juels. Optimistic mixing for exit-polls. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, pages 451–465, 2002. 40

[85] Jeremy Grace. External and absentee voting. *Challenging the Norms and Standards of Election Administration*, pages 35–57, 2007. 29

[86] Gurchetan S Grewal, Mark D Ryan, Liqun Chen, and Michael R Clarkson. Du-vote: Remote electronic voting with untrusted computers. In *2015 IEEE 28th Computer Security Foundations Symposium*, pages 155–169. IEEE, 2015. 35, 36

[87] Dimitris A. Gritzalis. *Secure Electronic Voting*, volume 7 of *Advances in Information Security*. Springer, 2003. 27, 28, 32, 33, 39, 41, 42, 43

[88] Jens Groth. Non-interactive zero-knowledge arguments for voting. In *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, pages 467–482, 2005. 42

[89] Jeff Grove. Acm statement on voting systems. *Commun. ACM*, 47(10):69–70, October 2004. 33

[90] Joshua Y Gwanshak and Amos Hyeladi. Contemporary issues of population census in nigeria. 2019. 95

[91] Stuart Haber, Josh Benaloh, and Shai Halevi. The helios e-voting demo for the iacr. 04 2019. 49

[92] Wei Han, Kefei Chen, and Dong Zheng. Receipt-freeness for groth e-voting schemes. *J. Inf. Sci. Eng.*, 25:517–530, 03 2009. 42

[93] S. Heiberg and J. Willemson. Verifiable internet voting in estonia. In *2014 6th International Conference on Electronic Voting: Verifying the Vote (EVOTE)*, pages 1–8, Oct 2014. 11, 48, 81, 83

[94] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, pages 539–556, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. 41, 42

[95] Jonathan T. Hiskey and Shaun Bowler. Local Context and Democratization in Mexico. *American Journal of Political Science*, 49(1):57–71, 2005. 70, 74, 95

[96] Sheng-Yu Hwang, Hsiang-An Wen, and Tzonelih Hwang. On the security enhancement for anonymous secure e-voting over computer network. *Computer Standards & Interfaces*, 27(2):163 – 168, 2005. 44, 45

[97] Kristine Höglund. Electoral violence in conflict-ridden societies: Concepts, causes, and consequences. *Terrorism and Political Violence*, 21(3):412–427, 2009. 131

[98] Democracy Report Internation. Electronic biometric verification of voters in pakistan: a silver bullet? resreport, March 2015. 131

[99] International Republican Institute (IRI). Election observation mission final report: Bangladesh parliamentary elections december 29, 2008. 2008. 93

[100] Victor Iwuoha. Ict and elections in nigeria: Rural dynamics of biometric voting technology adoption. *Africa Spectrum*, 53:89–113, 01 2018. 20, 131

[101] Markus Jakobsson. A practical mix. In *EUROCRYPT*, 1998. 38

[102] Markus Jakobsson. Flash mixing. In *Proceedings of the Eighteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '99, pages 83–89, New York, NY, USA, 1999. ACM. 38

[103] Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking. *IACR Cryptology ePrint Archive*, 2002:25, 2002. 38, 39, 120

[104] D. Jayasinghe, R. N. Akram, K. Markantonakis, K. Rantos, and K. Mayes. Enhancing emv online pin verification. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 808–817, Aug 2015. 109

[105] Danushka Jayasinghe, Raja Naeem Akram, Konstantinos Markantonakis, Konstantinos Rantos, and Keith Mayes. Enhancing EMV Online PIN Verification. In *IEEE TrustCom*, volume 1, pages 808–817, Helsinki, Finland, August 2015. 112

[106] Peter Sandholt Jensen and Mogens K. Justesen. Poverty and vote buying: Survey-based evidence from africa. *Electoral Studies*, 33:220 – 232, 2014. 35, 70

[107] Rui Joaquim, Paulo Ferreira, and Carlos Ribeiro. EVIV: an end-to-end verifiable internet voting system. *Computers & Security*, 32:170–191, 2013. 193

[108] J. Karro and J. Wang. Towards a practical, secure, and very large scale online election. In *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*, pages 161–169, Dec 1999. 31

[109] J. Karro and J. Wang. Towards a practical, secure, and very large scale online election. In *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*, pages 161–169, Dec 1999. 31, 32, 44

[110] A. E. Keshk and H. M. Abdul-Kader. Development of remotely secure e-voting system. In *2007 ITI 5th International Conference on Information and Communications Technology*, pages 235–243, Dec 2007. 42

[111] Reto E. Koenig, Philipp Locher, and Rolf Haenni. Attacking the verification code mechanism in the norwegian internet voting system. In James Heather, Steve Schneider, and Vanessa Teague, editors, *E-Voting and Identify*, pages 76–92, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 49

[112] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 27–40, May 2004. 29

[113] M. La Polla, F. Martinelli, and D. Sgandurra. A survey on security for mobile devices. *IEEE Communications Surveys Tutorials*, 15(1):446–471, First 2013. 133

[114] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security Privacy*, 9(3):49–51, May 2011. 76

[115] Fabrice Lehoucq. Electoral Fraud: Causes, Types, and Consequences. *Annual Review of Political Science*, 6(1):233–256, 2003. 70, 95

[116] Xiangxue Li, Dong Zheng, Kefei Chen, and Jianhua Li. Democratic group signatures with collective traceability. *Computers & Electrical Engineering*, 35(5):664–672, 2009. 46, 47

[117] Iuon-Chang Lin, Min-Shiang Hwang, and Chin-Chen Chang. Security enhancement for anonymous secure e-voting over a network. *Comput. Stand. Interfaces*, 25(2):131–139, May 2003. 44, 45

[118] Lukas Malina, Jan Smrz, Jan Hajny, and Kamil Vrba. Secure electronic voting based on group signatures. In *38th International Conference on Telecommunications and Signal Processing, TSP 2015, Prague, Czech Republic, July 9-11, 2015*, pages 6–10, 2015. 47, 48

[119] Lukas Malina, Arnau Vives-Guasch, Jordi Castellà-Roca, Alexandre Viejo, and Jan Hajny. Efficient group signatures for privacy-preserving vehicular networks. *Telecommunication Systems*, 58(4):293–311, 2015. 47

[120] Mark Manulis. Democratic group signatures: on an example of joint ventures. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2006, Taipei, Taiwan, March 21-24, 2006*, page 365, 2006. 46, 47

[121] Mark Manulis, Ahmad-Reza Sadeghi, and Jörg Schwenk. Linkable democratic group signatures. In *Information Security Practice and Experience, Second International Conference, ISPEC 2006, Hangzhou, China, April 11-14, 2006, Proceedings*, pages 187–201, 2006. 46, 47

[122] Konstantinos Markantonakis, Michael Tunstall, Gerhard P. Hancke, Ioannis G. Askoxylakis, and Keith Mayes. Attacking Smart Card Systems: Theory and Practice. *Inf. Sec. Techn. Report*, 14(2):46–56, 2009. 98

[123] Chuks Mba. Challenges of population census enumeration in africa: an illustration with the age-sex data of the gambia. *Research Review of the Institute of African Studies*, 20, 09 2005. 95

[124] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative instructions

and software model for isolated execution. In *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, pages 10:1–10:1, New York, NY, USA, 2013. ACM. 51, 67

[125] B. Meng. An internet voting protocol with receipt-free and coercion-resistant. In *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, pages 721–726, Oct 2007. 32, 43

[126] Rebecca T. Mercuri. Physical verifiability of computer systems. In *In International Computer Virus and Security Conference*, 1992. 29

[127] European Union Election Observation Mission. European union election observation mission to malawi -final report on the presidential and parliamentary elections, 2009. 2009. 130

[128] S. Mohanty and B. Majhi. A secure multi authority electronic voting protocol based on blind signature. In *2010 International Conference on Advances in Computer Engineering*, pages 271–273, June 2010. 45

[129] Yi Mu and Vijay Varadharajan. Anonymous secure e-voting over a network. In *14th Annual Computer Security Applications Conference (ACSAC 1998), 7-11 December 1998, Scottsdale, AZ, USA*, pages 293–299, 1998. 44, 45

[130] S. J. Murdoch, S. Drimer, R. Anderson, and M. Bond. Chip and PIN is Broken. In *IEEE Symposium on Security and Privacy*, pages 433–446, May 2010. 98

[131] Sujithra Muthuswamy and Padmavathi Ganapathi. Mobile device security: A survey on mobile device threats, vulnerabilities and their defensive mechanism. *International Journal of Computer Applications*, 56:24–29, 10 2012. 133

[132] C. Andrew Neff. A Verifiable Secret Shuffle and its Application to e-voting. In *Proceedings of the 8th ACM Conference on Computer and Communications Se-*

*curity (CCS)*, pages 116–125, Philadelphia, Pennsylvania, USA, November 2001. 39, 120

[133] Peter G. Neumann. Risks in computerized elections. *Commun. ACM*, 33(11):170, 1990. 29, 30

[134] Stephan Neumann and Melanie Volkamer. Civitas and the Real World: Problems and Solutions from a Practical Point of View. In *Seventh International Conference on Availability, Reliability and Security, ARES*, pages 180–185, Prague, Czech Republic, August 2012. 97

[135] SImeon Nichter. Vote Buying or Turnout Buying? Machine Politics and the Secret Ballot. *American Political Science Review*, 102:19–31, 2008. 74, 95

[136] Simeon Nichter. Conceptualizing Vote Buying. *Electoral Studies*, 35:315 – 327, 2014. 74, 95

[137] Chikodiri Nwangwu, Vincent Chidi Onah, and Otu Akanu Otu. Elixir of electoral fraud: The impact of digital technology on the 2015 general elections in nigeria. *Cogent Social Sciences*, 4(1):1549007, 2018. 20

[138] The Association of Electoral Administrators. It's time for urgent and positive government action. the aea's review of the 2017 local government elections and the uk parliamentary general election. Technical report, The Association of Electoral Administrators, September 2017. 72

[139] The Institution of Engineering and Technology. Internet voting in the uk: The issues, challenges and risks around internet voting. 2020. 19

[140] Cabinet Office. Cyber security strategy of the united kingdom: safety, security and resilience in cyber space. london: Crown copyright, 2009. 75

[141] Cabinet Office. The uk government's response to the electoral commission's reports on the 2017 uk parliamentary general election. Technical report, Cabinet Office, November 2018. 72

[142] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols, 5th International Workshop, Paris, France, April 7-9, 1997, Proceedings*, pages 25–35, 1997. 32, 43, 44

[143] Hakeem Onapajo. Violence and votes in nigeria: The dominance of incumbents in the use of violence to rig elections. *Africa Spectrum*, 49(2):27–51, 2014. 19, 130

[144] Yinyeh Ophelius and K A Gbolagade. Overview of biometric electronic voting system in ghana. 3, 07 2013. 131

[145] Haijun Pan, Edwin S. H. Hou, and Nirwan Ansari. E-NOTE: an e-voting system that ensures voter confidentiality and voting accuracy. In *Proceedings of IEEE International Conference on Communications, ICC 2012, Ottawa, ON, Canada, June 10-15, 2012*, pages 825–829, 2012. 47

[146] Haijun Pan, Edwin S. H. Hou, and Nirwan Ansari. RE-NOTE: an e-voting scheme based on ring signature and clash attack protection. In *2013 IEEE Global Communications Conference, GLOBECOM 2013, Atlanta, GA, USA, December 9-13, 2013*, pages 867–871, 2013. 28, 47

[147] Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 248–259, 1993. 38, 39

[148] K. Peng and F. Bao. Efficient proof of validity of votes in homomorphic e-voting. In *2010 Fourth International Conference on Network and System Security*, pages 17–23, Sep. 2010. 42

[149] Kun Peng, Riza Aditya, Colin Boyd, Ed Dawson, and Byoungcheon Lee. Multiplicative homomorphic e-voting. In *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, pages 61–72, 2004. 42

[150] Birgit Pfitzmann. Breaking efficient anonymous channel. In *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, pages 332–340, 1994. 39

[151] Sir Erick Pickles. Securing the ballot: Report of sir eric pickles' review into electoral fraud. Technical report, Department for Communities and Local Government, August 2016. 72

[152] Kim Ramchen and Vanessa Teague. Parallel shuffling and its application to prêt à voter. In *Proceedings of the 2010 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'10, pages 1–8, Berkeley, CA, USA, 2010. USENIX Association. 173

[153] H. Ran and W. Z. Peng. A protocol of electronic voting and blind digital signature based on elliptic curve. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 489–491, May 2011. 45

[154] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 552–565, 2001. 46

[155] F Rodríguez-Henríquez, Daniel Ortiz-Arroyo, and Claudia García-Zamora. Yet another improvement over the mu–varadharajan e-voting protocol. *Computer Standards & Interfaces*, 29:471–480, 05 2007. 44, 45

[156] Peter Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. PrÊt À voter: a voter-verifiable voting system. *Information Forensics and Security, IEEE Transactions on*, 4:662 – 673, 01 2010. 34

[157] Peter Y. A. Ryan. A variant of the chaum voter-verifiable scheme. In *Proceedings of the 2005 Workshop on Issues in the Theory of Security*, WITS '05, pages 81–88, New York, NY, USA, 2005. ACM. 28, 34

[158] Peter Y. A. Ryan and Steve A. Schneider. Prêt à voter with re-encryption mixes. In *Computer Security - ESORICS 2006, 11th European Symposium on Research in Computer Security, Hamburg, Germany, September 18-20, 2006, Proceedings*, pages 313–326, 2006. 33, 40, 173, 186

[159] Peter Y. A. Ryan, Steve A. Schneider, and Vanessa Teague. End-to-End Verifiability in Voting Systems, from Theory to Practice. *IEEE Security & Privacy*, 13(3):59–62, 2015. 17, 19, 21, 80, 97, 161, 173, 175, 180, 181, 182, 186, 188, 195

[160] Peter YA Ryan, Peter B Rønne, and Vincenzo Iovino. Selene: Voting with transparent verifiability and coercion-mitigation. In *International Conference on Financial Cryptography and Data Security*, pages 176–192. Springer, 2016. 36

[161] Bruce Schneier. Insiders - schneier on security. https://www.schneier.com/blog/archives/2009/02/insiders.html. Accessed: 2019-02-25. 74

[162] U. Serdult, M. Germann, F. Mendez, A. Portenier, and C. Wellig. Fifteen years of internet voting in switzerland [history, governance and use]. In *2015 Second International Conference on eDemocracy eGovernment (ICEDEG)*, pages 126–132, April 2015. 48

[163] Ida Sofie Gebhardt Stenerud and Christian Bull. When reality comes knocking norwegian experiences with verifiable electronic voting. In *5th International Conference on Electronic Voting 2012, (EVOTE 2012), Co-organized by the Council*

*of Europe, Gesellschaft für Informatik and E-Voting.CC, July 11-14, 2012, Castle Hofen, Bregenz, Austria*, pages 21–33, 2012. 49

[164] G. E. Suh, C. W. O'Donnell, and S. Devadas. Aegis: A single-chip secure processor. *IEEE Design Test of Computers*, 24(6):570–580, Nov 2007. 67

[165] Charles Taylor. Shared security, shared elections- best practices for the prevention of electoral violence. Technical report, American Friends Service Committee (AFSC), July 2018. 130

[166] Trustonic. Trustonic hardware security for mobile-derived credentials: with the kinibi trusted execution environment. Technical report, Trustonic, 2018. 51, 67

[167] EU Electoral Follow up Mission. Electoral follow-up mission to eu eom 2013 recommendations- efm kenya – final report. February 2016. 130

[168] Poorvi Vora. David chaum's voter verification using encrypted paper receipts. In *In DIMACS Workshop on Electronic Voting – Theory and Practice, Rugers*, 2005. 40

[169] H. Wei, Z. Dong, and C. Ke-fei. A receipt-free punch-hole ballot electronic voting scheme. In *2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, pages 355–360, Dec 2007. 42

[170] Doug Whiting, Russ Housley, and Niels Ferguson. Counter with CBC-MAC (CCM). RFC 3610, September 2003. 155

[171] Douglas Wikström. Five practical attacks for "optimistic mixing for exit-polls". In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography*, pages 160–174, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 40

[172] Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. Attacking the washington, d.c. internet voting system. In Angelos D. Keromytis, editor,

*Financial Cryptography and Data Security*, pages 114–128, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 76

[173] Peter Wolf, Adbul Alim, Brown Kasaro, Pontius Namugera, Mohammed Saneem, and Tamir Zorigt. Introducing biometric technology in elections. 35, 93, 95

[174] Zhe Xia, Chris Culnane, James Heather, Hugo Jonker, Peter Y. A. Ryan, Steve Schneider, and Sriramkrishnan Srinivasan. Versatile prêt à voter: Handling multiple election methods with a unified interface. In Guang Gong and Kishan Chand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010*, pages 98–114, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 173

[175] Zhe Xia, Steve A. Schneider, James Heather, and Jacques Traoré. Analysis, improvement, and simplification of prêt à voter with paillier encryption. In *2008 USENIX/ACCURATE Electronic Voting Workshop, EVT 2008, July 28-29, 2008, San Jose, CA, USA, Proceedings*, 2008. 34

# Appendix A

# Scyther Scripts

*This appendix lists the scyther script and verification result for our mutual authentication voting protocol*

## A.1 Introduction

We present the scyther script for the generic mutual authentication voting protocol in chapter 5 to show authentication of the voter is not compromised and hence voter's eligibility is verified

Listing A.1: Scyther Code

```
hashfunction h;
usertype SessionKey;
usertype tag;
const BiometricVerified, PinVerified: tag;
usertype Auth;
usertype Data;
secret Cert: Function;
macro IDvc = VC;
//macro IDvcSigned = {IDvc}sk(IA);
```

```
protocol protPK(VC,VT,IA) {

 role VT {
   fresh nvt: Nonce;
   fresh AuthData: Auth;
   var IDvc: Data;
   var nvc: Nonce;
   var cnt: Data;
   var K:  SessionKey;

   send_1 (VT,VC, nvt,Cert(VT));
   recv_2(VC,VT, {h( IDvc,nvt,nvc)}sk(VC),
     {IDvc,nvc,K}pk(VT),Cert(VC));
   send_3(VT,VC, {h(AuthData,nvc)}sk(VT),
     {AuthData,nvc}K);

   claim_a1(VT, SKR, K);
   claim_a2(VT, Secret, AuthData);
   claim_a3(VT, Niagree);
   claim_a4(VT, Nisynch);
 }

 role VC {
   fresh nvc, nvc1: Nonce;
   fresh IDvc: Data;
   fresh K:  SessionKey;
   var nvt: Nonce;
```

```
  var niss: Nonce;
  var AuthData: Auth;
  const cnt: Data;

  recv_1(VT,VC, nvt,Cert(VT));
  send_2(VC,VT, {h(IDvc, nvt,nvc)}sk(VC),
    {IDvc, nvc,K}pk(VT),Cert(VC));
  recv_3(VT,VC, {h(AuthData,nvc)}sk(VT),
    {AuthData,nvc}K);
  claim(VC,Running,IA,IDvc,
    nvc1,BiometricVerified,PinVerified,
    k(VC,IA));
  send_4(VC,IA,
    h(IDvc,nvc1,BiometricVerified,
    PinVerified,
    k(VC,IA)));
  recv_5(IA,VC, h(niss,nvc1, k(VC,IA)));
  claim(VC,Commit,IA, niss,nvc1, k(VC,IA));

  claim_b1(VC, Alive);
  claim_b2(VC, SKR, K);
  claim_b3(VC, Secret, AuthData);
  claim_b4(VC, Niagree);
  claim_b5(VC, Nisynch);
}

role IA {
  fresh niss: Nonce;
```

```
    var nvc1: Nonce;
    var IDvc: Data;

    recv_4(VC,IA, h(IDvc,nvc1,
      BiometricVerified,
      PinVerified, k(VC,IA)));
    claim(IA,Running,VC,niss,nvc1, k(VC,IA));
    send_5(IA,VC, h(niss,nvc1, k(VC,IA)));

    claim(IA,Commit,VC,IDvc,nvc1,
      BiometricVerified,
      PinVerified, k(VC,IA));
  }

}
```

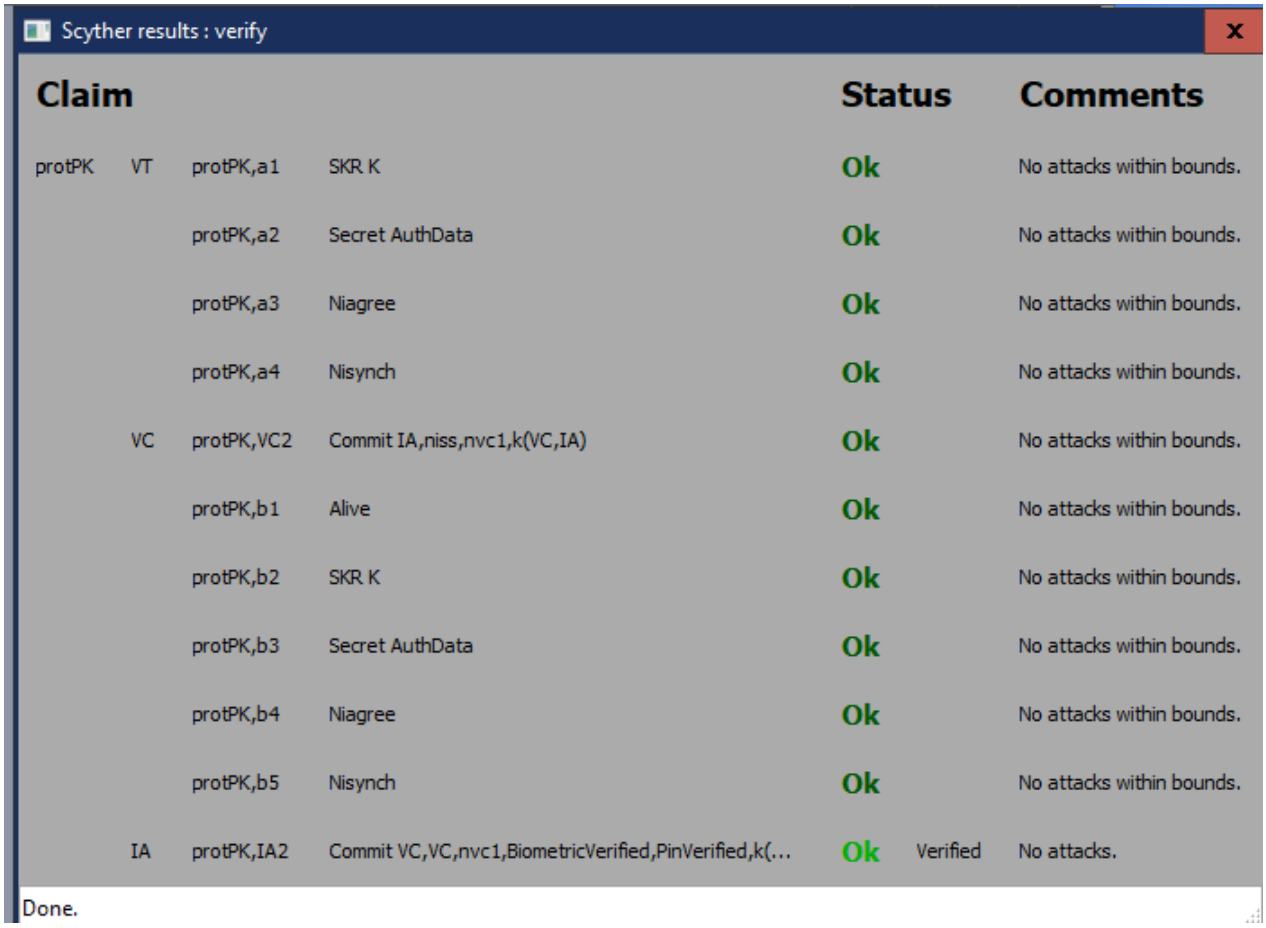


Figure A.1: Mutual Authentication Voting Scheme-Scyther Verification