Challenging Environments: Using Mobile Devices for Security

Submitted by

Sheila Cobourne

for the degree of Doctor of Philosophy

of the

Royal Holloway, University of London

2018

Declaration

I, Sheila Cobourne, 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.....(Sheila Cobourne)
Date:

In memory of
John Boulton 1954-2011
Dorothy Boulton 1930-2016

Abstract

The advent of the Internet, and advances in computing and phone technology have transformed the way society interacts and conducts business. There are well established security processes and protocols that exist to protect people's privacy and the sensitive credentials needed for secure transactions. However, many areas of the world do not have access to the high-quality technical infrastructure, equipment and expertise necessary for these security procedures to be effective. These are challenging environments, and this thesis examines how mobile devices can be used to enhance the security of applications in them. Three application areas are investigated: remote e-voting; m-payments; and authentication. Two of these areas are then investigated in the (differently challenged) online Virtual World (VW) environment.

Eight use cases are presented in total, employing a range of features available on mobile phones to address identified security issues. The main contributions can be found in solutions that introduce security through a Smart Card Web Server (SCWS) application installed on the tamper-resistant smart card chip Subscriber Identity Module (SIM) found in a mobile device. These solutions include remote e-voting on a mobile device, branchless banking and offline Single Sign-On authentication. The use of well-established and standardised security protocols with tamper-resistant hardware enhances the security of these proposals, and distributing processing to a number of SIMs protects against attacks such as Distributed Denial of Service. Other work describes a Bitcoin SMS m-payment scheme, and preliminary investigations into the potential for using gesture recognition dynamic biometrics on a mobile phone. The VW applications, log-in authentication and in-world voting, are also outlined. All proposals are analysed (informally and formally if appropriate) with respect to defined security requirements. A discussion of the security and practicality of SCWS solutions is given, along with suggested future research directions.

Acknowledgement

People say it takes a village to raise a child: a PhD thesis also requires the help of a great many people.

The first person I would like to thank is, of course, my supervisor Professor Keith Mayes, Founder Director of the Smart Card Centre (SCC) and Head of the School of Mathematics and Information Security at Royal Holloway, University of London, who has been a constant source of inspiration throughout my PhD. Without his encouragement and support I would not have had the opportunity to achieve my long-held dream of gaining a doctorate. It has been a privilege to work as his student.

Thanks must also go to other members of the SCC, Professor Konstantinos Markantonakis and Dr Raja Naeem Akram.

Fellow students in the SCC have shared the highs and lows of my PhD journey, with much laughter, brilliance and patience. So thank you Lazaros Kyrillidis, Graham Hili, Benoit Ducray, Danushka Jayasinghe, Assad Umar, Mehari Msgna, Hafizah Mansor and Sarah AbuGhazalah. I couldn't have done it without you.

I am truly grateful to the WISDOM group as it has allowed me to further the cause of Women in STEM in the company of some amazing people, especially Thyla Van Der Merwe.

I am thankful that my friends continued to remind me that I was actually doing OK when the pressures of life as a mature student threatened to overwhelm me, and the unconditional support of my husband Mike and my newly-extended family has been precious beyond words.

Thank you all.

Author's Publications

Research Papers

Remote e-Voting

1. L. Kyrillidis, S. Cobourne, K. Mayes, S. Dong, and K. Markantonakis, "Distributed e-voting using the Smart Card Web Server", in *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS2012)*, IEEE, 2012, pp. 1–8.[1]

2. S. Cobourne, L. Kyrillidis, K. Mayes, and K. Markantonakis, "Remote e-Voting Using the Smart Card Web Server", *International Journal of Secure Software Engineering (IJSSE)* vol. 5, no 1, pp.39–60, 2014.[2]

m-Payment

1. S. Cobourne, K. Mayes, and K. Markantonakis, "Using the Smart Card Web Server in Secure Branchless Banking", in *International Conference on Network and System Security (NSS2013)*, Springer, 2013, pp. 250–263.[3]

2. D. Jayasinghe, S. Cobourne, K. Markantonakis, R.N. Akram, and K. Mayes, " Philanthropy On The Blockchain", in *11th WISTP International Conference on Information Security Theory and Practice (WISTP2017)*, 2017. [4]

Authentication

1. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Authentication Based on a Changeable Biometric using Gesture Recognition with the Kinect™",

in *2015 International Conference on Biometrics (ICB2015)*, IEEE 2015 pp. 38–45. [5]

2. L. Kyrillidis, S. Cobourne, K. Mayes, and K. Markantonakis, "A Smart Card Web Server in the Web of Things," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys 2016)*, Springer, 2016, pp. 769-784 [6]

3. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Comparison of Dynamic Biometric Security Characteristics against other Biometrics", in *2017 IEEE International Conference on Communications (ICC2017)*. [7]

4. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Gesture Recognition Implemented on a Personal Limited Device", in *8th International Conference on Information and Communication Systems (ICICS2017)* [8] (Nominated for Best Paper)

Virtual Worlds

1. L. Kyrillidis, G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Virtual World Authentication using the Smart Card Web Server", in *International Symposium on Security in Computing and Communication' (ISSCC2013)*, pp. 30–41. [9]

2. S. Cobourne, G. Hili, K. Mayes, and K. Markantonakis, "Avatar Voting in Virtual Worlds", in *5th International Conference on Information and Communication Systems (ICICS2014)*, pp. 1–6.[10] (Nominated for Best Paper)

3. G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Practical Attacks on Virtual Worlds", in *International Conference on Risks and Security of Internet and Systems (CRiSIS2014)*, pp. 180–195. [11]

Book Chapters

1. A. Tomlinson, and S. Cobourne, "Chapter 6: Security for Video Broadcasting", *Smart Cards, Tokens, Security and Applications, 2nd Edition*, K.E. Mayes, and

K. Markantonakis, (Eds.), (2017), Springer. [12]

2. J. Cadonau, D. Jayasinghe, and S. Cobourne, "Chapter 11: OTA and Secure SIM Lifecycle Management", *Smart Cards, Tokens, Security and Applications, 2nd Edition*, K.E. Mayes, and K. Markantonakis, (Eds.), (2017), Springer. [13]

Articles

1. K. Mayes, S. Cobourne, and K. Markantonakis, "Near Field Technology in Challenging Environments", *Smart Card Technology International. NFC and Contactless (2011)*, p. 65-69. [14]

Contents

List of Figures

List of Tables

List of Notation

E_K(Z)	Encryption of data Z with key K
H(Z)	Hash of data Z
ID_X	Identity of entity X
N_X	Random Nonce generated by entity X
$NAME_X$	Name of entity X, (i.e. a short identifying text)
Ph_X	Phone number of entity X
PK_X/ SK_X	Public/ Secret Key pair of entity X
S_X/ V_X	Signing/ Verification key pair of entity X
$x\|\|y$	the concatenation of x and y
x, y	shorthand for $x\|\|y$
X	Entity X
(Z)$Sign_K$	Signature on data Z with signature key K

List of Abbreviations

2FA	Two Factor Authentication
3G	Third Generation
ACP	Access Control Policy
ACPE	ACP Enforcer
ANN	Artificial Neural Network
API	Application Program Interface
BIP	Bearer Independent Protocol
BTC	Bitcoin Currency
CA	Certificate Authority
CES	Consumer Electronics Show
DDoS	Distributed Denial of Service
DRE	Direct-Recording Electronic
DTW	Dynamic Time Warping
ECDSA	Elliptic Curve Digital Signature Algorithm
EER	Equal Error Rate
FAP	Full Administration Protocol
FAR	False Acceptance Rate
FPR	False Positive Rate
FxOS	Firefox OS
GDP	Gross Domestic Product
GPS	Global Positioning System
GSM	Global System for Mobile communication)
HCE	Host Card Emulation
HMM	Hidden Markov Model
HOTP	Hash-based One-Time Password
HTML	Hyper Text Mark-up Language
HTTP	Hypertext Transfer Protocol
HTTPs	HTTP over SSL or HTTP Secure
IMEI	International Mobile Equipment Identity
IVR	Interactive Voice Response
KDC	Key Distribution Centre
KYC	Know Your Customer
LAP	Lightweight Administration Protocol
LDAP	Lightweight Directory Access Protocol

MITM	Man in The Middle
MMORPG	Massively Multiplayer Online Role-Playing Game
MMS	Multimedia Messaging Service
MNO	Mobile Network Operator
NFC	Near Field Communication
OMA	Open Mobile Alliance
OTA	Over the Air
OTP	One-Time Password
OWASP	Open Web Application Security Project
P2SH	Pay to Script Hash
PAV	Prêt à Voter E-Voting System
PGD	Pretty Good Democracy
PIC	Personal Identification Code
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PP	Protection Profile
PSK-TLS	Transport Layer Security Pre-Shared Key
QR	Quick Response
RAS	Remote Administration Server
RFID	Radio Frequency IDentification
RSK	Rootstock
RW	Real World
SBTC	Smart Bitcoin Currency
SCWS	Smart Card Web Server
SE	Secure Element
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SL	Second Life
SMS	Short Message Service
SMSC	Short Message Service Center
SQL	Structured Query Language
SSL	Secure Socket Layer
SSO	Single Sign On
TAR	True Acceptance Rate
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
TOTP	Time-based One-Time Password
TPR	True Positive Rate
TRR	True Rejection Rate
TSL	Trusted Secure Layer
TSM	Trusted Service Manager
TTP	Trusted Third Party
UICC	Universal Integrated Circuit Card
UID	Unique Identifier (VW entity)
USB	Universal Serial Bus

USIM	Universal Subscriber Identity Module
USSD	Unstructured Supplementary Service Data
VCA	Vote Counting Application
VCL	Vote Code List
VFS	Vote Forwarding Server
VSS	Vote Storage Server
VW	Virtual World
WAP	Wireless Application Protocol
WOW	World of Warcraft
XSS	Cross-site Scripting

Part I

Introduction and Background

Chapter 1

Introduction

Contents

This chapter explains the motivation behind this thesis, its research questions and objectives, and the methodology used to achieve these objectives. The chapter then lists the contributions and publications that have resulted from the work undertaken.

1.1 Motivation

The advent of the Internet and advances in computing and phone technology have transformed the way society interacts and conducts business. There are well established security processes and protocols that exist to protect people's privacy and the sensitive credentials needed for their secure transactions. However, many areas of the world do not have access to the high-quality technical infrastructure, equipment and expertise necessary for these security procedures to be effective.

Regions of extreme poverty, places with insufficient governance, hazardous geographical areas, war zones, or in humanitarian crisis scenarios (e.g. after natural disasters such as earthquakes and tsunamis) may not have the physical infrastructure necessary for these security measures to work effectively. For example, there may be inadequate, damaged or destroyed power supplies, communications and transport, with poor physical security for equipment and personnel. Social conditions may be difficult: for example, in a mass exodus of individuals in a humanitarian crisis, formal identity documents and/or a home address may be lost, hindering identification and authentication processes. Furthermore, low literacy levels, untrusted local institutions and untrained officials may result in established security procedures not being followed correctly. In such situations it is a demanding task to implement fundamental security principles in systems which deal with sensitive and potentially life-saving information. Security of information in these challenging environments will be the focus of the research presented in this thesis[1].

Basic security requirements of confidentiality, integrity and availability need to be met by any system, whatever the environment. Security questions arise which are harder to solve in these challenging conditions than in more well resourced areas. For example, how can you be sure that the person you are transacting with is not an imposter when online verification of identity may not be possible? If communication is difficult, how can sensitive information be protected locally in a secure and tamper-resistant manner until connectivity is restored?

Mobile communications can have a real impact in these situations. Mobile phones are easy to use, portable and secure with a range of options for communication, and the more advanced handsets (smart phones) have sophisticated processing capabilities which previously would have required the power of a laptop to execute. According to the ITU Telecommunication Bureau in a 2016 report [23], seven billion people (95% of the global population) live in a area that is covered by a cellular network, and the use of mobile phones within the GSM network coverage is considerably higher compared

[1]It should be noted, that the challenging environments described in this thesis can also be found in the developed world, in under-served communities (known as the "digital divide" [22].)

Figure 1.1: "Sahal Gure Mohamed, 62, Texts on his Mobile Phone" by Internews Europe is licensed under CC BY 2.0. Accessed 21 April 2017. https://www.flickr.com/photos/internewseurope/7887050306

to other communication technologies - 53% of the world's population is not using the Internet. Figure 1.1 illustrates the far reaching availability of mobile phones. Mobile phones and the cellular network might be the only available communications option, but this has provided a "leap-frog" technology that can bring services to otherwise under-served communities. The next section discusses how mobile phones can be used to enhance security in these environments.

1.2 Using Mobile Devices for Security

Systems need a back office infrastructure, and to maintain high security levels this is normally in a trusted environment with protected servers and trained security-aware staff. In a challenging environment, however, this secure back-office function may be difficult to provide locally. Mobile phones can provide links to secure servers in other parts of the world, so a secure back office infrastructure residing in a trusted environment far removed from the operational environment can be used. The trustworthiness, security and acceptability of a system is therefore enhanced.

The mobile phone itself has a range of available sensors, functions and secure storage options that can be utilised in applications. For example, fingerprint readers can be used for biometric authentication such as Apple's TouchID [24] and Samsung's fingerprint verification [25]. Alternatively, the mobile phone could be used in two-factor authentication schemes, as the phone is "something you have" and a user PIN is "some-

thing you know" e.g. [26].

Often, the Short Message Service (SMS) can be used as a second channel to send authentication messages containing One Time Passwords (OTPs). Applications such as m-banking e.g. [27] or authentication software e.g. [28] can be installed on the mobile device, but the phone platform is generally regarded as untrusted as it can be tampered with by the user ("jail-broken" or "rooted") or be infected with malware.

Of most relevance to this research, a mobile phone also contains the most widely available smart card in existence, the mobile phone Subscriber Identity Module (SIM). Smart cards are designed with tamper-resistant chip technology and storage, along with specialised protocols and security algorithms that enable information to be processed securely [29]. Modern smartphones may also contain a similar tamper-resistant chip known as the Secure Element (SE). Applications installed in the tamper resistant SIM environment that use the tightly managed, standardised protocols and functionality of the Smart Card Web Server (SCWS) will form the main focus of this thesis.

The next section defines research questions that this work will consider along with corresponding research objectives.

1.3 Research Questions and Objectives

With this background in mind, research questions have been formulated. These are:

1. **RQ-1:** How can introducing a trustworthy infrastructure based on mobile devices and/or alternative methods of authentication address the differing security requirements of a range of use cases in these environments?

2. **RQ-2:** How can a trusted element in a mobile device be used to enhance security in challenging environments, where there may be limited access to technical infrastructure and resources?

These research questions give rise to main and secondary objectives for the work presented in this thesis. They are:

1. **RO-1:** (Main Objective) Design security solutions using mobile devices to enhance security in a range of use cases in challenging environments.

2. **RO-2:** (Secondary Objective) Design improved authentication methods that can be used in challenging environments.

The next section describes the methodology adopted to meet the stated research objectives.

1.4 Methodology

The methodology adopted in this thesis started with a literature review that was conducted to identify how mobile devices/ SIMs/ mobile networks have been used in existing schemes in challenging environments. The literature review was then used to derive selection criteria and provide the rationale for choosing three application areas that were subsequently to illustrate representative security problems. These are:

- **Remote e-Voting:** this was chosen because the use of mobile devices for this application area has not been widely studied, and mobile e-voting could provide real benefits for remote communities, or in societies where attendance at a polling station could be dangerous because of the potential for election-related violence;
- **m-Payment:** this was selected because there are many schemes in the developing world designed to provide financial inclusion for "unbanked" communities[2], but serious security problems have been identified with some of these solutions e.g. [27];
- **Authentication:** this was included as it is a service that underpins all secure solutions, and investigating alternative methods of local authentication could assist in situations where online connectivity or reliable identification credentials are not available.

Within each application area, two use cases were studied in detail, and solutions were proposed. These proposals were subjected to informal security analysis, and when appropriate, a mechanical formal security analysis using the automated tool Scyther [30] was also done. Then, as an aside to the main work, two of these application areas (e-voting and authentication) were additionally investigated in a Virtual World (VW) environment, which exhibits similar security challenges to resource-poor Real World (RW) scenarios. VWs and their security issues are described in detail in Section 2.7. The full list of use cases along with the research questions/ research objectives they address is shown in Table 1.1.

1.5 Contributions

The work presented in this thesis has made several main contributions by using tamper-resistant trusted hardware, via SCWS applications on a SIM card. The work is novel

[2]In the poorest communities of the world, where average income is less than $5 a day, many people do not have access to safe, secure and affordable financial services that could help them climb out of poverty: these are referred to as "unbanked" individuals.

Table 1.1: Use Cases vs Research Questions and Objectives

Ref	Use Case Description	RQ1	RQ2	RO1	RO2
	Remote e-Voting				
EV-1	SCWS Voting using Prêt à Voter (PAV)	✓	✓	✓	✓
EV-2	SCWS Voting using Estonian I-Voting	✓	✓	✓	✓
	m-Payment				
MP-1	SCWS Branchless Banking	✓	✓	✓	✓
MP-2	Bitcoin SMS m-Payments	✓	×	✓	×
	Authentication				
Auth-1	Offline SCWS Single Sign-On	✓	✓	✓	✓
Auth-2	Gesture Recognition Biometric	✓	×	✓	✓
	Virtual World Applications				
VW-1	SCWS Online VW Log-In	✓	✓	✓	✓
VW-2	VW Voting	✓	×	✓	×

because to the author's knowledge, SCWS applications have not been proposed as secure solutions before.

Secondary contributions have also been made in authentication and VW applications.

Main Contributions:

1. Using the SCWS to provide tamper resistance and protection against Distributed Denial of Service (DDoS) attacks in remote e-voting; illustrated by using e-Voting systems Prêt à Voter and Estonian I-voting as examples [1, 2];

2. Using advanced SIM capabilities - including the SCWS - to provide security improvements on existing SIM-based m-Payment schemes [3];

3. Enabling access to secure blockchain technology via an SMS m-payment system, to be used for charitable donation provisioning in offline humanitarian aid scenarios [4];

4. Providing secure authentication in an offline environment through a local Single Sign-On procedure via SCWS chips installed in a SIM and standalone security module [6];

Secondary Contributions:

1. Investigation into the use of gesture recognition to provide a dynamic, two-factor one-step biometric authentication method [5, 7, 8];

2. Improved online log-in to VWs using authentication via the SCWS, geolocation and One Time Password (OTP) processes [9]

3. Introduced privacy for in-world VW voting, by locating sensitive e-voting processing in a Trusted Secure Layer external to the VW, and using the mobile phone network as second channel for security information [10].

Several peer-reviewed conference papers have been published following this research, and they are listed at the end of this chapter.

1.6 Thesis Outline

The thesis is organised into parts as follows.

Part 1: Introduction and Background provides the context of the research, discusses previous work and gives background information about the application areas selected for further study i.e. remote e-voting, m-payment and authentication. Chapter 2 presents a literature review, background to the selected application areas and the rationale behind their choice. The VW environment is also discussed. Chapter 3 explains the technologies used in solutions put forward in this thesis, in particular, the SCWS.

Part 2: Application Areas and Use Cases proposes solution(s) that can meet security requirements for each use case shown in Table 1.1. Chapter 4 covers remote e-voting use cases *EV-1* and *EV-2*. Chapter 5 deals with mobile payment use cases *MP-1* and *MP-2*. Chapter 6 investigates alternative authentication methods through the use cases *Auth-1* and *Auth-2*. Chapter 7 describes solutions for two of these application areas in a VW setting, through use cases *VW-1* and *VW-2*.

Part 3: Analysis and Conclusion provides further analysis of the previously presented work. Chapter 8 expands upon the security of the SCWS, and the use case solutions presented previously. SCWS implementation issues are then discussed. Chapter 9 draws the thesis to its conclusion by assessing how well the stated research objectives have been met and suggesting future research directions.

Appendices include additional information that supports the research in this thesis. Appendix A contains supplementary information relevant to some of the use cases. Appendix B describes the formal analysis tool Scyther, which was used to verify the

security of some of the solutions presented. The Scyther scripts used and the results obtained are shown here.

1.7 Chapter Summary

This chapter outlined the motivations for the research, the research objectives and methodology, and showed the contributions which were made as a result of the work undertaken. The structure of the thesis was laid out. The author's related peer-reviewed publications are now listed.

1.8 Related Publications

Publications are listed by application area.

Remote e-Voting

1. L. Kyrillidis, S. Cobourne, K. Mayes, S. Dong, and K. Markantonakis, "Distributed e-Voting using the Smart Card Web Server", in *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS2012)*, IEEE, 2012, pp. 1–8.[1]

2. S. Cobourne, L. Kyrillidis, K. Mayes, and K. Markantonakis, "Remote e-Voting Using the Smart Card Web Server", *International Journal of Secure Software Engineering (IJSSE)* vol. 5, no 1, pp.39–60, 2014.[2]

m-Payment

1. S. Cobourne, K. Mayes, and K. Markantonakis, "Using the Smart Card Web Server in Secure Branchless Banking", in *International Conference on Network and System Security (NSS2013)*, Springer, 2013, pp. 250–263.[3]

2. D. Jayasinghe, S. Cobourne, K. Markantonakis, R.N. Akram, and K. Mayes, " Philanthropy On The Blockchain", in *11th WISTP International Conference on Information Security Theory and Practice (WISTP2017)*, 2017. [4]

Authentication

1. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Authentication Based on a Changeable biometric using Gesture Recognition with the Kinect™", in *2015 International Conference on Biometrics (ICB2015)*, IEEE 2015 pp. 38–45. [5]

2. L. Kyrillidis, S. Cobourne, K. Mayes, and K. Markantonakis, "A Smart Card Web Server in the Web of Things," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys 2016)*, Springer, 2016, pp. 769-784 [6]

3. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Comparison of Dynamic Biometric Security Characteristics against other Biometrics", in *2017 IEEE International Conference on Communications (ICC2017)*. [7]

4. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Gesture Recognition Implemented on a Personal Limited Device", in *8th International Conference on Information and Communication Systems (ICICS2017*) [8] (Nominated for Best Paper)

Virtual Worlds

1. L. Kyrillidis, G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Virtual World Authentication using the Smart Card Web Server", in *International Symposium on Security in Computing and Communication' (ISSCC2013)*, pp. 30–41. [9]

2. S. Cobourne, G. Hili, K. Mayes, and K. Markantonakis, "Avatar Voting in Virtual Worlds", in *5th International Conference on Information and Communication Systems (ICICS2014)*, pp. 1–6.[10] (Nominated for Best Paper)

3. G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Practical Attacks on Virtual Worlds", in *International Conference on Risks and Security of Internet and Systems (CRiSIS2014)*, pp. 180–195. [11]

Chapter 2

Background

Contents

Background information describing how mobile devices and their applications are currently used in challenging environments is provided. Three application areas for further research are selected (Remote e-Voting, m-Payment and Authentication) and the rationale for their choice is supported by a review of the security of existing schemes. This is followed by a description of the Virtual World environment, and further study in this (differently) challenging area is then identified.

2.1 Challenging Environments

Examples of challenging environments can be seen in war zones, post-disaster regions, or communities characterised by extreme poverty and low literacy levels: information security may be affected by human, technical or societal factors. For example, individuals may not have formal documents, making identification and authentication difficult. Special measures may be needed, such as the use of different Know Your Customer (KYC) banking rules for financial aid distribution to displaced populations in humanitarian crises [31]. Physical and technical infrastructures may be damaged or non-existent, leading to unreliable communication capabilities [32], and available equipment may have technical constraints such as reliance on battery power (illustrated in Figure 2.1), limited storage capability and old versions of hardware/ software [33]. Societal factors may include reduced expectations of individual privacy, especially if devices are shared[1]. Low literacy and language issues may be encountered [35, 36] which will reduce the effectiveness of technical systems. Additionally, bribery, collusion, coercion and corruption may be the societal norm: examples can be seen at `http://afrobarometer.org`, where one study showed that in Kenya, police, government officials, Members of Parliament, and business executives are most widely perceived as corrupt [37]. As seen previously, mobile communications can provide a "leap-frog" technology that can bring services with economic and social benefits to these communities. The next section discusses some of the projects and commercial schemes that use mobiles in this way.

2.2 Mobile Devices in Challenging Environments

There are a number of initiatives that use mobiles in developing countries. For example, the Mobile for Development team within the GSMA shows the following project areas on its website, all of which utilise mobiles to bring financial and societal advantages: Connected Society; Mobile Money; Digital Identity; Connected Women; Mobile for Development Utilities; m-Agri; m-Health; and Disaster Response [38]. Some example schemes are now described.

In *M-health* applications, mobiles can be used to disseminate medical information and collect data about the health of communities and individuals e.g. [39, 40]. Mobile data collection project examples include: monitoring treatment during a pneumonia epidemic in Pakistan [41]; obtaining sanitation information from villages after the Haiti

[1] In 2012, an ICT household survey carried out in Kenya found that 25% of low income people (at the so-called "Bottom of the Pyramid") who owned a mobile phone shared it with a family member, usually the spouse [34].

Figure 2.1: "Mobile Phone Charging Station" by Adam Cohn is licensed under CC BY-NC-ND 2.0 Accessed 26 July 2017. https://www.flickr.com/photos/adamcohn/6311096617/

earthquake (Smart Bucket) [42, 43]; and following a dengue fever outbreak in Mexico [44]. Other humanitarian projects have focused on binding individual identification to health records. For example, in one scheme [45, 46, 47], patients are given Radio Frequency ID (RFID) tags as ID tokens, healthcare staff are given a mobile RFID read/write device (such as a phone with Near Field Communication capability) and the health care centre uses the data collected in an electronic medical record system.

In *m-Payment* schemes SMS messages can be used for transactions that allow the unbanked access to secure financial services and hence remove the physical security issues of handling cash itself. The most well known m-payment service in the world is M-PESA (see Figure 2.3), operated by Safaricom in Kenya [48], and there are many similar schemes implemented in other countries e.g. [49, 50, 51, 52, 53]. SMS messaging has been used in humanitarian aid programmes e.g. [54, 55, 56]: in one example, a relief operation in Syria used vouchers for aid items that were sent direct to individuals via SMS [57].

Alternative uses for mobile phones can be seen in Delay Tolerant Network implementations such as ByteWalla [58, 59, 60, 61, 62] where the phone is used as a "data mule" to address infrastructure challenges by providing asynchronous Internet access [2] . Another proposal, Serval mesh networks [64, 65], provides peer-to-peer connectiv-

[2]Delay Tolerant Networking as defined by the IETF has a specific bundle layer on top of the standard OSI layers, described in the Bundle Protocol Specification (RFC 5050) [63]. In this thesis, it is assumed that although network outages can occur, they are not permanent and connectivity will be restored at some stage. Thus the use of these specialised data bundles is not needed in the solutions presented.

ity in offline environments by using a mobile phone application with mesh extender hardware to increase the available communication range.

The use of a PIN with a mobile phone can help identify and authenticate an individual. Research has shown that illiterate users can remember and use strings of numbers to access services. For example when using prepayment meters, illiterate users could manage to input twenty-digit sequences (as long as they were arranged as five groups of four digits) [66]. Also, it was found that semi-literate users were comfortable typing digits but could not locate symbols on a phone [67].

Lack of identification introduces a major barrier to accessing basic services, however: for example, proof-of-identity is mandatory to register a mobile SIM; also, to open a mobile money account KYC regulations must be met. It has been reported that 20% of adults attribute their lack of identification as a reason for being unable to access financial services [68]. Biometric applications (i.e. "something you are/do") can be used for authentication when other credentials are unavailable, but there are associated privacy concerns [69].

The next section discusses the availability and capability of the phones often found in challenging environments.

2.2.1 Mobile Phone Availability

In the poorest communities, basic "feature phones" are commonly available, but these have text-based interfaces that are not always suitable for the end-user e.g. illiterate users [35]. A number of initiatives have aimed to bring low-cost smart phones to developing regions e.g. [70, 71, 20, 72]. Some of the phones which have been promoted in this way are shown in Table 2.1. The Android One programme in India [73] takes a different approach: the handsets are relatively expensive (around 100$) but they run stock versions of Android[3] and are guaranteed to receive the latest OS updates directly from Google for two years after launch, increasing the security of the platform. Following on from Android One, Google announced Android Go [74] at the Mobile World Congress 2018 [75]. Android Go is a stripped-down version of Android Oreo, and has been designed to run on inexpensive (i.e. under 100$), low-end devices with limited internet connectivity. In contrast to the Android One programme, any company can use Android Go in their products (rather than enter into partnership with Google) which could increase the availability of low-cost smartphones generally [76, 77]. The Mozilla Firefox OS (FxOS) [78] was designed to work with very low specification handsets and

[3]At time of writing - June 2017 - Android One phones shipped with Android 7.0 (Nougat)

Table 2.1: Low Cost Smartphones [20, 21]

Phone	**Karbonn Smart A50S**	**Micromax Bolt A27**	**Spice Smart Flo Edge**	**NOKIA ASHA 503**	**MTN Steppa**	**Intex Cloud FX**	**Alcatel Klif**
OS	Android 4.2.2	Android 2.3.5	Android 2.3.5	Nokia Asha 1.2	Android 2.3.5	Firefox OS 1.3	Firefox OS 2.0
RAM	256MB	256MB	256MB	128MB	512MB	128 MB	256MB
Bluetooth	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Price US$	$45	$48	$46	$46	$48	$33	$40
Release Date	June 2014	Jan 2013	Sept 2013	Nov 2013	Jan 2014	July 2014	Q2 2015

limited data connectivity[4]. There were more expensive FxOS devices with NFC capability e.g. Alcatel Fire S, the Fx0 in Japan and the Mozilla Flame reference phone [79], but FxOS support for SE processing was never made available. Neither Android One nor the phones shown in Table 2.1 have NFC capability, although Bluetooth is usually available. A screenshot from an FxOS phone (i.e. the user interface of an Alcatel Flame [79]) is shown in Figure 2.2.

According to a 2017 GSMA report [80], there are approximately 4 billion smartphone connections globally, but smartphone usage varies greatly across regions. For example, in mid-2017, Eastern Africa and South Asia had smartphone adoption levels of 25% and 30% respectively, low compared to the global average (over 50%). The GSMA attributes a major contributing factor for this is the high rate of poverty in these areas. The GSMA also note that in India an average priced smartphone can cost up to 16% of income for poor and low-income groups, and as a result they estimate that over 134 million people in India cannot afford even one of the cheapest smartphones. Even though it is predicted that smartphone prices will decrease in these emerging markets, they will still be out of reach for these underserved communities in the near future.

As smartphones are not necessarily available in the environments being studied, several of the solutions presented in this thesis aim to be usable on low specification equipment.

[4]FxOS was an open-source web-based mobile platform applied on top of Mozilla's Linux Boot-to-Gecko operating system, with a security model including sandboxing apps, a permissions scheme and API access controlled by an application runtime layer: Mozilla discontinued all work on the FxOS operating system in September 2016. The smartphone part of the project is now entirely maintained by Mozilla's volunteer community, and branded as B2G OS.

2.3 Application Areas to be Studied

The methodology adopted in this research is to study representative application areas as use cases, that span a wide range of security challenges. Suitable solutions that use mobile devices to enhance security will then be proposed. Each application area will be investigated in enough detail to allow its particular security issues to be understood and solutions devised: exhaustive analysis is not the purpose of this thesis. In order to give as broad an overview as possible, three application areas have been chosen: a new theme for research, Remote e-voting using a mobile; a topic with many existing implementations, m-payment, and an enabling technology, authentication. As an aside to the main work, two of these applications (remote e-voting and authentication) are also explored in the (differently) challenging VW environment.

The next sections give background information about how mobiles are currently used in these application areas. The rationale behind the choice of each application area is also given.

2.4 Remote e-Voting

2.4.1 Background - Remote e-Voting

Elections are important democratic events, and traditionally, voting is performed in person at controlled physical centres i.e. poll-sites. It can be a challenge to engage citizens and encourage them to vote, especially if voters are immobile or geographi-

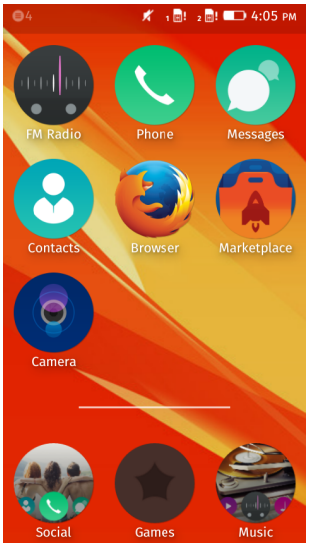

Figure 2.2: Firefox OS User Interface

cally remote. Elections have fundamental security requirements that votes should be recorded as cast, counted as recorded and not linked to a specific voter. Only eligible voters should be allowed to vote, and they can only cast one vote each [81]. Electronic voting (e-voting) uses electronic processes for one or more of the following tasks in an election: voter identification, vote casting, and/or vote counting. No entities in an e-voting system are considered to be trustworthy, and the stringent and often contradictory security requirements must be met to ensure the election's integrity remains intact.

Some e-voting systems are designed to address voting security requirements in the controlled environment of an election poll-site. Examples include fully electronic systems such as Votebox [82], Direct Recording Electronic (DRE) machines [83, 84]; paper-based ballots such as Prêt à Voter [85] and the Scratch Card voting system [86].

Remote e-voting enables a voter to cast their vote over the Internet. Participation could be improved by using remote e-voting systems, as a voter can use their own computer or mobile device to cast their vote. Examples of practical implementations of remote e-voting include elections in Estonia [87] and Switzerland [88]. The number of potential adversaries is very much higher for remote e-voting systems compared to paper-based poll-site voting, and a successful attack could have far-reaching implications e.g. state-level actors may have a vested interest in affecting the outcome of an election.

Although many e-voting processes can be cryptographically protected to ensure the integrity and confidentiality of the votes cast, Rivest [89] identified a critical problem with remote implementations, i.e. "interfacing the voter to the cryptography". Security weaknesses in hardware, operating systems and software mean that equipment cannot be trusted, so for example, the voter's equipment could be infected with malware that tampers with the vote. This is known as *"the secure platform problem"*. Several methods to address this have been proposed [90]. These include: having a "clean" operating system and voting application; using special hardware attached to a PC; secure PC operating systems i.e. trusted computing; test ballots; and security by obscurity. Code sheet voting is also popular, when voting authorisation codes are sent to voters before the election, via a second channel such as the postal service: examples here include Pretty Good Democracy [18], the work of Helbach et al. [91, 92] and Randell and Ryan [86].

Remote e-voting systems could be attacked using Internet vulnerabilities to disrupt an election. For example, there have been a number of security concerns about the Estonian Internet voting system, both in overall design and technical implementation. For example, there are no DDoS countermeasures, it may be possible to link a voter

to a vote and the procedures for cancelling re-votes may impact accuracy [93]. In the 2011 elections, there were technical web server and browser problems which hindered the voting process [94]. It was also reported that there was an application that could change the contents of the vote on the user's PC without them knowing, although this complaint was not upheld by the voting authority [95]. The scheme is not voter-verifiable, i.e. the voter has no way of checking whether their vote has been counted as cast: however, the voter has the option to cast a paper vote at a later stage if they have any doubts about the security of the I-voting system[5].

Technical attacks on remote e-voting infrastructures and associated sites have been reported:

- DDoS attacks against centralised voting web-servers were seen in the 2010 Washington D.C. election [96] and the 2012 Canadian New Democratic Party Elections [97]. In the Washington D.C. case, the e-voting system was broken into within 48 hours of it becoming available, and by taking control of the election server, the attackers "changed every vote and revealed almost every secret ballot" [96].

- Remote e-voting systems that have implemented anti-DDoS measures have opened up new routes for attack. The 2017 state election of Western Australia (WA) used an Internet voting system (I-Vote) from third-party vendor Scytl, in conjunction with Imperva Incapsula, a content delivery network which provides a DDoS mitigation service by operating as a TLS proxy. It was found that the I-Vote server had been misconfigured, and JavaScript performed by the DDoS protection service could be used maliciously to compromise voter credentials and modify ballots [98].

- There were also reports of Russian influence in the US elections in 2016 [99] and a possible DDoS attack on the U.K.'s Referendum voter registration site [100].

- In 2014, an online (unofficial) democracy polling site `https://popvote.hk/` that was canvassing opinion on future Hong Kong elections was subjected to a large and sophisticated DDoS attack [101].

Trust in an electoral process may be low [102], and violence can occur. For example, Kenya has a history of corruption and systemic abuse of office by public officials, and every election since 1991 has resulted in violence [103]. The violence that erupted after the 2008 elections was widespread and prolonged. Also, in the Russian elections in 2011,

[5]The I-voting scheme was amended to include voter-verifiability after the 2015 Estonian elections [19].

there were several cyber-attacks, and individuals posted videos of ballot-box stuffing on social media. Rumours of election-rigging circulated on the Internet - fuelled by the fact that in some areas voter turnout appeared to exceed 140%. Protesters clashed with armed police and 300 activists were detained: on a later occasion another 2,000 protesters were dispersed by riot police [104].

Using Mobile devices in Remote e-Voting

There are currently very few examples of mobile phone based e-voting systems: one example of an e-voting scheme that has been implemented as an application on a mobile devices is SEAS [105]. This was formally analysed by Campanelli et al. [106]. However, mobile voting applications are vulnerable because mobile phone operating systems/ applications cannot be trusted to perform correctly (the secure platform problem).

Scytl developed a telephone voting system that uses a standard land line or mobile phone [107]. However Scytl e-voting systems have been criticised in the past, notably their claims of end-to-end verifiability [108], to which they responded by claiming the report was inaccurate [109]. It was a Scytl system that was the victim of the DDoS attack in the 2012 NDP elections [97], and another of their systems was used in the 2017 state election of Western Australia (WA) mentioned above.

The latest version of Scytl's e-voting software uses client-side JavaScript, which has been tested on Android and iPhone browsers as well as desktop implementations [110], therefore it can be used for mobile voting. However, a vulnerability was found in the JavaScript voting client that Scytl implemented for the State General Elections 2015 of New South Wales [111]. This occurred because third party code[6] was included for monitoring purposes. However, the 3rd party server that hosted the code had the FREAK [112] vulnerability present, so it would be possible to exploit this and tamper with the voting client code in the voter's browser to modify the vote. Scytl's view was that this vulnerability's potential damage to vote integrity was akin to malware on the voter's device [110] - which brings us back to the secure platform problem.

2.4.2 Rationale for Studying e-Voting

Remote e-voting on a mobile device is a relatively new area to research, and the systems that have been implemented have security issues. As shown above, there are two interesting security aspects that warrant further investigation, i.e. DDoS protection and methods for overcoming the secure platform problem. Processing votes in the tamper-resistant environment of the SIM in the mobile device would help with both these

[6]Scytl does not recommend including third party code from external servers.

security problems. Designing a solution that offers an offline mobile voting capability could be especially useful in situations when attending a poll-site in person may be dangerous or impractical, and the communication infrastructure is not reliable. A mobile device can also also become an authentication factor ("something you have") in the voting process.

Remote e-voting proposals are presented in Chapter 4.

2.5 Mobile Payments

2.5.1 Background - M-Payment

Mobile financial systems are used to make payments to individuals/merchants. In [27], the authors describe different types of mobile money system[7].

- **Mobile Payments:** mobile device uses traditional banking infrastructure e.g. Apple Pay [113] as an interface for existing accounts;
- **Mobile Wallets:** these store payment credentials for multiple services. Some m-payment schemes also have mobile wallets e.g. PayPal [114];
- **Branchless Banking systems:** these often have simple enrolment procedures, do not necessarily rely on Internet connectivity but will use SMS, Unstructured Supplementary Service Data (USSD)[8], or Interactive Voice Response (IVR) in transactions. However, the security of USSD and SMS is known to be weak [117, 118].

2.5.2 M-Payment Schemes

m-Payment schemes can run using SMS-based transactions, mobile applications or NFC applications in conjunction with a Secure Element (SE) on the mobile device.

SMS-based

SMS messaging is the lowest common denominator of mobile device communication, available on all models including basic feature phones. SMS-based m-payment schemes can be used with low-specification phones to providing financial services to unbanked users. However, SMS-based m-payment schemes may not be appropriate if customers are illiterate, unfamiliar with technology and unable to access conventional text-based

[7] In this thesis, the term "m-Payment" will refer to all types of mobile money systems.

[8] USSD is a standardised mechanism [115, 116] that allows a mobile device to communicate directly with an MNO application to obtain MNO-specific supplementary services.

Figure 2.3: M-PESA: "M-PESA Mobile Money Transfer in Kenya" by Erict19 is licensed under CC BY-NC 2.0, accessed 23 May 2017. https://www.flickr.com/photos/54869669@N02/6940221629/

user interfaces [36], and in this case it is a challenge to provide services securely. Using a smart phone would give a more intuitive and user-friendly experience, but in general the high cost of suitable handsets has precluded this.

Examples of SMS m-payment schemes include:

- **M-PESA:** The most successful m-payment service in the developing world is M-PESA, operated by Safaricom in Kenya [48]. This provides SMS funds transfers and cash transactions using a network of authorised agents. (The text-based user interface is shown in Figure 2.3.)

- **Bitcoin SMS Schemes:** There are schemes which carry out Bitcoin transactions using SMS text messages e.g. Coinapult [119], BTC Wallet [120], Coinbase SMS service [121][9]. Most of these schemes involve initial online access to set up individual Bitcoin Wallets, which can then be maintained via SMS as well as via online transactions. Bitwala offers a Bitcoin remittance transfer service to Uganda, Tanzania and Nigeria that uses mobile money services [122, 123], but again, the user needs online access to their Bitcoin Wallet. Attempts to integrate Bitcoin directly with M-PESA have not been totally successful due to business

[9]The Coinbase SMS service was discontinued in March 2017, in favour of smartphone apps

pressures [124, 125, 126]. (Other proposals to carry out Bitcoin transactions using mobile phones need a smartphone app to interact with online Bitcoin wallets e.g. BTC Wallet [120].)

- **SMS as Second Channel** The SMS service can also be used as a second channel for authentication messages containing OTPs or other security credentials from the bank/ financial organisation[10].

There are security issues associated with the use of SMS messaging for confidential/ authentication data in SMS-based m-Payment schemes. Many use proprietary security mechanisms (security by obscurity), and there are well documented security issues in GSM/3G mobile networks (e.g. [128]). The SMS service operates on a "best effort" basis: messages can be delayed, dropped or arrive out of order. SMS messages therefore cannot be considered confidential, and as encryption is not applied to the service by default, messages can be intercepted, snooped and spoofed and the SMS service is vulnerable to man-in-the-middle attacks [129]. An attack on M-PESA involved spoofed bank-originating SMS messages (along with knowledge of a secret obtained by social engineering) and caused a security breach which defrauded an agent of 35,000 Kenyan Shillings [130]. Malware can intercept/ suppress SMS messages [131], point to phishing websites and specifically target banking applications, e.g. mobile Zeus trojan [132]. An integral part of the SMS system is the Short Message Service Centre (SMSC) where messages are stored before delivery to the intended recipient. Messages are stored in plaintext, so they could be tampered with by insiders at the SMSC. Possible attacks include replay attacks, or Denial of Service (DoS) attacks where a large number of repeated SMS messages overload the system: fuzzing attacks such as the "SMS of Death" can result in an unusable device [133]. Other attack methods include interception/redirection using false base stations in GSM networks, and SS7 hacking [134].

Applications on Mobile Device

An application on the mobile phone could connect a customer to a bank's web server. Here, authentication credentials are sent over the Internet to the bank for checking; SMS OTP codes are sometimes used as an added security measure. However, transmitted customer credentials can be attacked: and the bank's web server is exposed to all standard Internet security threats, e.g. the Open Web Application Security Project (OWASP) Top Ten [135], or DDoS attacks [136]. Sensitive (i.e. high-value) information

[10]However, in May 2016, the National Institute of Standards and Technology (NIST) published a guideline recommending the deprecation of SMS authentication as a second factor for strong authentication [127].

is stored on phone handsets, forming attractive targets for malware. The risk of banking trojans has grown year on year. For example, in 2017, Trend Micro observed a 94% increase in banking malware samples over those analysed in 2016: they described the latest malware as "more obfuscated, persistent, and flexible" [137]. Macafee had similar findings, of over 16 million mobile malware occurrences in the third quarter of 2017, nearly double that of 12 months previously [138]. Infection can occur via Multimedia Messaging Service (MMS) messages or Bluetooth connections e.g. Commwarrior [139]. The phone is therefore not regarded as a trustworthy platform.

A comprehensive analysis of branchless banking mobile applications can be seen in [27], where the authors found a wide range of security problems. They encountered:

> *"systemic vulnerabilities spanning botched certification validation, do-it-yourself cryptography, and myriad other forms of information leakage that allow an attacker to impersonate legitimate users, modify transactions in flight, and steal financial records."*

This report also found that often the terms of service for branchless banking applications shifted the liability for these security problems to the customer.

Near Field Communication Solutions

Near Field Communication (NFC) is a standardised short-range wireless technology[11]. NFC allows a phone to behave as a contactless smart card or a contactless smart card reader, over short distances (hence the term Near Field). Data can be transferred over distances of up to 20 cm if both devices produce their own radio field (active mode), or if only one device generates a radio field the operating distance is less, up to 10 cm (passive mode). A core part of the NFC framework is the Security Element (SE) that is designed to provide trust between users and Service Providers (SPs). Sensitive information can be securely stored and processed on this tamper-resistant hardware, and the placement of the SE in the mobile device is determined through ownership, trust and management interests. The SE can be integrated within the SIM, implemented as an embedded hardware module or installed on a Secure Memory token.

There are many existing schemes which use NFC for m-payments, with account credentials stored on an SE e.g. Apple Pay [113]: some schemes use Host Card Emulation (HCE) in software for payment processing. However, as phones with NFC capability are not commonly available in the environments being studied (see the low-cost smartphone programmes shown in Table 2.1), NFC-based payment infrastructures will not

[11] Relevant standards are ISO/IEC 18092 [140],ISO/IEC 14443 [141], ISO/IEC 15693 [142], ISO/IEC 21481 [143] and Sony Felica [144]

be included in the solutions presented in this thesis, and are considered out of scope.

2.5.3 Rationale for Studying M-Payments

As shown above, there are many m-payment solutions, but there are also associated security concerns, often due to weaknesses in the underlying mobile communications network. This application area was chosen to see if systems could be proposed that would be suitable for use on basic feature phones that would result in security improvements on existing SMS-based m-payment schemes.

The proposals for m-Payment solutions are presented in Chapter 5.

2.6 Authentication

2.6.1 Background - Authentication

There are two challenges which are of interest with regard to authentication: lack of identification credentials, and a potentially unreliable communication infrastructure that both reduce the options for the technical solutions that can be offered.

Identification Credentials

In the absence of reliable identification credentials, a mobile phone can be used for two-factor authentication [26] i.e. "something you have" as well as "something you know" if a PIN is used to access the phone: a PIN can be used even if the user has low literacy skills as described in Section 2.2.1. Mobile software applications can be used to provide authentication services, for example the Blizzard Authenticator [28] for the VW World of Warcraft (WOW) [145]. Phone cameras can record authentication details: this could be in the form of 3D barcodes (Quick Response) QR codes [146, 147] for mobile payment applications [148, 149]; 2D barcodes in m-commerce [150]; or photos of documents that can be used in mobile transactions [151]. However, mobile phone operating systems are increasingly becoming targets for malware [137, 138], so storing credentials in a tamper-resistant device, such as the SIM or SE, is an attractive option that could be used to authenticate to other services e.g. [152, 153].

A phone's microphone/ speaker can transmit audible authentication details: for example a "sonic barcode" data-over-audio toolkit from Chirp [154], which is used in MNO solutions and video games such as Activision Blizzard's Skylanders Imaginators [155]; and Tagpay from TagExpress which uses their patented Near Sound Data Transfer (NSDT^{TM}) for authentication [156]. These schemes have the advantage that

they can be easily used with feature phones, but the disadvantage is that these examples are all proprietary solutions.

A biometric approach could also be used if other identification is not available: there are large scale biometric projects that aim to give individuals a unique identity, to be stored on a national database for authentication. A prominent example here is the Aadhaar project in India [157], where people enrol by providing minimal demographic information along with a captured biometric (fingerprint/iris). A random and unique 12 digit identity number (the Aadhaar number) is then generated which can be used in future authentication situations.

Alternatively, sensors on a mobile phone can capture biometric information about a person. Smartphones can be used to capture inherent information about the user: an individual's particular pattern of mobile phone use can provide behavioural biometric input to authentication systems, for example through keystroke dynamics [158] or touchscreen dynamics [159]. More advanced smartphones with in-built fingerprint readers are becoming commonplace and can be used as a biometric capture devices e.g. unlocking a phone via fingerprint readers such as Apple's TouchID [24] and Samsung's fingerprint verification [25]. Other examples include MasterCard's "pay-by-selfie" service that uses the camera [160]; voice authentication [50]; or continuous authentication schemes which use the behavioural characteristics of the user as they interact with their phone during normal use, by utilising the phone's accelerometer data [161]. A promising research area is that of using a mobile phone's on-board accelerometer to record specific gestures which can then be used as a dynamic biometric, to provide the advantage of one-step two factor authentication by combining a knowledge factor (i.e. the gesture itself) with a biometric (i.e. the individual movement needed to perform the gesture e.g. [162]).

Online and Offline Authentication

There are several online authentication schemes which use mobile phones: these include optical challenge-response procedures such as 2-clickAuth [163], and authentication involving SMS messages, discussed in detail in [164]. In the work "The Quest to Replace Passwords" [165] the authors identified and assessed the usability, deployability and security of a range of authentication methods, including mobile phone authentication: their examples were Phoolproof [166]; Cronto [167]; MP-Auth [168]; OTP over SMS; Google 2-Step Account Verification. They found that mobile phone authentication solutions were generally at least as good, and often better, than passwords over a range of security criteria (summarised in Table 1 in [165]). However, mobile authentication systems that can be used in offline scenarios are not easily found.

2.6.2 Rationale for Studying Authentication

Authentication underpins secure transactions and access to information. However, authentication of individuals may be difficult to achieve in offline situations, or where identity credentials are not available. Investigating alternative methods that a mobile device could use to provide local authentication (either through the use of the tamper-resistant SIM, or its installed sensors) would be of interest in disconnected environments, with the aim to achieve a security level equivalent to a 4-digit PIN. If available, a smartphone would provide a larger range of available sensors.

The proposals for alternative authentication methods are discussed in Chapter 6. Please note: the use of identity cards and passports for authentication will be considered out of scope for this thesis.

The next section describes other environments which have similar inherent security issues to those found in the resource-poor challenging environments described previously. These are the fully online environments that form Virtual Worlds (VWs)[12]. Background information about VWs and their identified security threats will now be given.

2.7 Virtual World Applications

2.7.1 Background - Virtual Worlds

Virtual Worlds are highly popular, specialised online environments where people can interact in real-time via digital beings known as avatars[13]. Figure 2.4 shows a screenshot of a VW with several avatars. The term "Virtual World" has been defined as "a synchronous, persistent network of people, represented as avatars, facilitated by networked computers" [170]. Three main types of VW have been identified [171]:

- ludic worlds (game worlds/ MMORPGS), where the objective is to complete quests and enhance your avatar's skills and reputation, usually as part of a "guild" or community (e.g. Blizzard's World of Warcraft (WOW) [145]);
- civic worlds which aim to mimic real life as much as possible, with features such as commerce, meeting places, democracy and education (e.g. Linden Research's Second Life (SL) [172]); these are self-contained "social spaces" [173];

[12]Certain VWs can also be referred to as MMORPGs - Massively Multi-Player Online Role-Playing Games.

[13]The term "avatar" was first used by Morningstar [169] to describe a digital representation of a user in a VW, so for the purposes of this thesis an avatar will be regarded as a human entity.

- social worlds which are akin to graphical social networks (e.g. Kaneva [174][14] and SmallWorlds [175]); these tend to complement existing RW friendships and civic participation.

Figure 2.4: Virtual World Avatars: "Avatars of Second Life Unite" by Torley is licensed under CC BY-SA 2.0, accessed 13 June 2017. https://www.flickr.com/photos/torley/16032382936

VWs have their own virtual currencies, which can be used to buy and sell virtual goods: for example, in SL the currency is Linden Dollars, in WOW the currency is Gold. As a result, in-world economies have evolved, and individuals and business organisations have been able to exploit their money-making potential. In 2015, it was reported that the Gross Domestic Product (GDP) of SL was $500 Million [176], and that users were cashing out approximately $60 million per year [177]: in 2017, it was calculated that SL content creators were making more money from the VW than its actual developer, Linden Labs [178]! There is the potential for RW financial gain, so malicious and fraudulent activities can also be seen in VWs.

There are VW-specific security threats which affect entities within the VW itself, and these have have been formalised in Lee and Warren's Virtual World Security Threat Matrix [179] (later updated in [180]), and by ENISA [181]. Lee and Warren identify threats such as:

- information exchanged between avatars may not be encrypted and is not confidential/ private; legal protection such as the EU Privacy Directive [182] only applies to natural persons i.e. the avatars' RW controllers;

- avatar activities may be monitored; VW developers use sophisticated data mining and behaviour analysis to detect in-world cheating, through techniques very

[14] At the time of writing (2017) the Kaneva website now only shows casino-style "social 3D games".

similar to spyware [181][15];

- avatar identity cannot necessarily be verified, and identity can be reverse engineered or stolen through social engineering;
- avatars can be attacked, hijacked or stalked. Some VWs, such as Eve Online [185], actively encourage their users to devise new methods of scamming, deceiving, or attacking each other, as part of the gameplay. Eve Online has also had issues with corrupt VW developers [186];
- scripted bots (e.g. copybots), malicious applications and objects can crash virtual locations, seize control of avatars and disrupt VW events in VW-specific denial of service attacks;
- an avatar cannot determine if a VW object they are interacting with is genuine;
- cheating by users to gain advantage with respect to honest users of the VW; for example, there is also a form of cheating called "gold farming", where RW sweat shops employ workers to play games specifically to generate artefacts and skilled avatars which can be sold for hard cash on the RW black market [187];
- harassment ("griefing") which restrict the activities of VW avatars, including damaging VW locations and property (it has been reported that SL griefing is turning into a "full-blown crime" [188]);
- fraudulent virtual financial activities; for example, in Eve Online, a user set up a bank, took a great deal of money as deposits, and then disappeared with the proceeds leaving virtual investors out of pocket [189]. There is very little legal redress in the case of virtual theft, as most VW developers retain intellectual property rights over in-world items created by users so no actual theft is deemed to have taken place [190] [191];
- malicious servers which harvest personal information for future illegal use;
- and attacks through the VW Client, since once downloaded, the VW developers have no more control over it, and users can tamper with its software by changing application logic or incoming/outgoing data. For example, malware may be

[15]Reports leaked by Edward Snowden in December 2013 revealed that UK and US Government Security Agencies (GCHQ and National Security Agency (NSA) infiltrated WOW and SL as they believed that terrorist or criminal networks could use the anonymity of the VW environment to communicate secretly or launder money [183]. One leaked document revealed "while GCHQ was testing its ability to spy on Second Life in real time, British intelligence officers vacuumed up three days' worth of Second Life chat, instant message and financial transaction data, totaling 176,677 lines of data, which included the content of the communications" [184].

present on the user's machine [192], or the user's machine can be compromised following VW identity theft [193].

Identification and authentication procedures for users of VWs can be fairly limited, mostly relying on static username/password combinations, which are easily compromised. Virtual goods and identities can be stolen if an account is hacked, with the danger that the user's computer can then be compromised and RW identity theft could occur [193]. Organised crime syndicates have been known to use the anonymity of VWs to hide money laundering and other illegal activities [194, 195].

A study into VW user experience showed that 22.9%+ of male players and 32%+ of female players had told secrets to their VW friends which they had not revealed to anyone in the RW [196], thus exhibiting a very relaxed attitude to information disclosure. The perceived anonymity of the VW environment can also lead to poor security awareness [181]. In the RW, there are basic security steps an individual can take to check if an action is likely to be insecure (e.g. not clicking on links sent from unknown sources, checking the SSL padlock on a browser): there are no clear VW equivalent measures [197].

Mobile Devices and VWs

As a VW is a fully online, persistent environment, RW devices such as mobile phones do not immediately spring to mind as appropriate security enablers. However, there is an overlap between virtual and real worlds. There are many VW services that are conducted externally to the world itself. These can include: VW-specific forums such as the SL Community [198]; more generalised MMORPG discussion spaces e.g. [199]; official VW object marketplaces that use RW payment systems e.g. Blizzard's Online Shop where you can buy virtual pets or give them as gifts e.g. [200]; or official marketplaces where trades are negotiated in the RW but paid for in the VW using virtual currency such as Eve Online's marketplace forum e.g. [201] .

Examples of this RW/VW overlap can be seen in SL, where there was a facility to use an in-world telephone system called AvaLine. This allowed users to make phone calls to/from the RW whilst appearing as their avatar persona [202]: communication with a non-SL user was also possible from within the VW. This was a very popular service - in 2009, over 15 billion minutes of voice services were used [203] - but the facility has now been discontinued.

There are existing VW security services that use phones. For example, the Battle.net Authenticator is either a physical token or an application on supported mobile devices, used for two-factor authentication to protect against unauthorised account ac-

cess [28]. The VW developers advise using this in conjunction with "SMS Protect" [204] for text message verification of account recovery/ suspicious login attempt/ password or security feature changes. However, there is a reported security vulnerability in the Battle.net mobile authenticator application [205]. It is vulnerable to a passive eavesdropper during the initialisation process due to a weak one time pad key generation algorithm on the client side, so Man-in-the-Middle attacks are possible. (Details can be seen in [206])

2.7.2 Rationale for Studying VW Applications

Of the previously selected application areas, m-Payment does not apply to the VW environment as its payment infrastructure is based either on standard external RW e-commerce facilities such as PayPal, or VW currency that supports the in-world economy. However, weak authentication can lead to identity theft and other RW problems for users of VWs, so investigating methods for enhancing VW authentication would be useful. Equally, the remote e-voting application could be applicable in VWs: voting in-world would enhance the immersive experience for VW users. An interesting area to study would be how to maintain the privacy of the vote in an environment where all activities are monitored: a mobile phone that receives voting information via a second communication channel (the MNO network) would be a helpful security option.

Proposed solutions for VW Applications are discussed in Chapter 7.

2.8 Chapter Summary

This chapter described RW challenging environments, and presented background information relating to the application areas chosen for further study in this thesis. The applications chosen are: a new research area (remote e-voting on mobile phones); a well established area (mobile payments); and enabling technologies for secure processes (authentication). The challenging VW environment was also described, and two of the application were selected for further study in VWs.

Chapter 3

Technology Background

Contents

This chapter covers background information about technologies that are used in the research solutions presented later in the thesis. Smart cards are designed with tamper-resistant chip technology and storage, with specialised protocols and security algorithms, so that information can be processed securely. The mobile phone Subscriber Identity Module (SIM) is the most widely available smart card in existence. This chapter describes the SIM and its standardisation, and this is followed by an explanation of the features, communications and security of the Smart Card Web Server (SCWS) which provides web server functionality implemented in the restricted environment of the SIM.

3.1 Smart Cards in Mobile Telecommunication

Smart cards are designed with tamper-resistant chip technology and storage, with specialised protocols and security algorithms, so that information can be processed securely [29]. A brief description of the use of smart cards in telecoms will now be given, based on work from several sources [29, 207, 208, 209].

3.1.1 UICCs and SIMs

The UICC, often known as the Universal Integrated Circuit Card[1] is a physically secure chip inserted into a mobile device that uses smart card technology to identify a user, their services (and their billing plan) to their Mobile Network Operator (MNO). The UICC contains a microprocessor along with its own data storage and software, with operator-defined profiles installed during manufacture that enable identification and authentication with the mobile network.

The Subscriber Identity Module (SIM) was the term that was originally used to describe both the physical card smart card in a Global System for Mobile Communications (GSM) phone (2G) and the telecommunication application software on that ran on the smart card. When it became possible for the SIM to operate alongside other smart card applications on the physical card (e.g. payment, travel or loyalty applications), i.e. when Universal Mobile Telecommunications System (UMTS) networks were introduced (now referred to as 3G), the physical hardware/low level software platform for telecommunications became known as the UICC. The 2G SIM application evolved into a 3G telecommunication application software that was termed the USIM.

The UICC can have multiple applications on it: for example it can have both SIM and USIM applications so the widest range of handsets can access a particular MNO's services. In this thesis, the term "SIM" will be used as a generic term to describe the smart card that consists of the UICC and its telecommunication access application software i.e. USIM (for 3G networks) or SIM (for GSM networks). Modern SIMs can have advanced features such as the Smart Card Web Server (SCWS) [15] which introduces web server functionality to the SIM environment and provides a rich interface for the user. Advanced SIMs can also perform public-key processing (PKI-capable SIM), using standardised cryptographic algorithms for encryption/ digital signatures [210, 211].

The SIM Application Toolkit (SAT or STK) is an interface between the SIM card and the handset that includes a set of commands used to build applications: for example, the SIM card can send short messages, set up a call or display menu and text items. STK applications often employ a text based menu approach and can provide

[1] The GSMA [208] states that UICC is neither an abbreviation nor an acronym.

simple user interfaces. Application updates can be delivered over-the-air (OTA) [13]. It is possible to run services such as banking applications/transactions securely on the SIM card, as seen in M-PESA [212], using encryption.

There are many standardisation bodies for telecommunications smart cards, which have helped to establish the SIM as the most widely available smart card in the world. The most relevant ones are now briefly described.

3.1.2 Standards

The UICC conforms to standards written by the European Telecommunications Standards Institute, (ETSI) and its Smart Card Platforms (SCP) group [213].

Inter alia, ETSI SCP define and maintain UICC standards for:

- two types of access (ISO and USB);
- Secure Remote Management procedures;
- an Application Programming Interface (API) set.

The custodians of the GSM specifications are The Third Generation Partnership Project (3GPP) [214], and they manage and maintain 3G/4G and the SIM/ USIM specifications (amongst others). The ISO7816 series of standards are particularly important for SIMs. The first four standards cover physical/electrical aspects, protocols and inter-industry commands [215, 216, 217, 218]. The full list of ISO7816 standards is available from the International Organization for Standardization website [219].

The STK interface is specified in GSM specifications: 3GPP TS 11.14 for SIM [220] and 3GPP TS 31.111 for USIM [221]. These specifications do not define how STK commands can be used interoperably on SIM cards from different SIM vendors. To accommodate this, a number of other standards exist that specify how to access STK functionality and create interoperable Java SIM applications: for example, 3GPP TS 03.19 (for SIM) [222]/ 3GPP TS 31.130 for USIM [223], 3GPP TS 03.48 [224], and GlobalPlatform Card Specifications [225].

Of particular relevance to the work in this thesis, specifications on the features and security for the SCWS are produced by the Open Mobile Alliance (OMA) [226]. Using web server functionality in the SIM environment forms the basis of several of the proposals in this thesis, so the SCWS will now be described in more detail.

3.2 The Smart Card Web Server (SCWS)

The Smart Card Web Server (SCWS) is an HTTP 1.1 server [227] implemented in the tamper resistant environment of the SIM: its features and functionality are standardised by the OMA. The OMA Specification states

> *"The SCWS is a web server, running within the Smart Card, to which local HTTP applications in the device can connect. The security considerations are the same as with any remote server that the user can browse with the handset Web browser. The SCWS shall implement HTTP and HTTPS and thus provide the same level of authentication, confidentiality and integrity as provided by other Web servers."* [15].

When it was first introduced, the SCWS was designed to provide MNOs and developers with a powerful execution environment that combined the advantages of both local runtimes (access to device APIs, local storage) and development using web languages (easy development, ability to retrieve data from the internet).

Additionally, the SIMAlliance proposed that the SCWS could be combined with contactless technology interfaces embedded in a single SE so access is fast and always on, even offline. This would improve security as transactions are made directly between the SCWS and the contactless SE [228].

The capabilities of the SCWS (version 1.2) will now be briefly outlined, summarised from the following OMA specifications [15][2]:

- Smartcard-Web-Server Approved Version 1.2.1 13 Sep 2013
- Smartcard Web Server Enabler Architecture Approved Version 1.2 05 Mar 2013
- Enabler Release Definition for Smartcard-Web-Server Approved Version1.2.1 13 Sep 2013
- Smartcard Web Server Requirements Approved Version 1.2 05 Mar 2013

3.2.1 SCWS Features

The purpose of the SCWS is to serve web pages locally to the handset's browser. These pages are either static HTML pages or dynamically created by Java applets running inside the SIM i.e. it serves both static and dynamic content through the use of on-card applications. The SCWS is owned and operated by the MNO, and is only accessible from authorised applications on the phone handset (based on an Access Control Policy

[2] A more detailed explanation of the security and management of the SCWS can be found in [229].

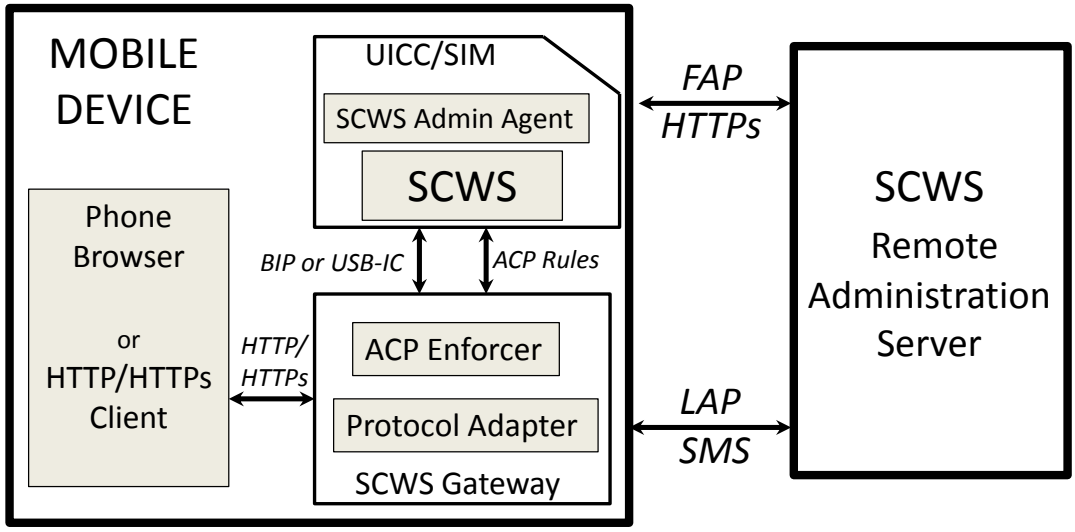

Figure 3.1: Smart Card Web Server Architecture (adapted from [15])

(ACP)) or a trusted Remote Administration Server (RAS) controlled by the MNO or an authorised third party (e.g. a bank). The SCWS should not be accessed by any other entity [15].

As the SCWS is a lightweight web server that has been specially designed to operate in a resource restricted environment like a SIM card, the OMA specification states that it is expected that the SCWS must implement a minimal set of HTTP 1.1 features. These are: GET; HEAD; POST; PUT; DELETE; and optionally OPTIONS, TRACE and CONNECT. The SCWS must support Basic Authentication and may support Digest Authentication as defined in IETF RFC 2617 [230], but an application can implement its own authentication scheme such as an application specific user name and password/PIN. It is possible for users to change their SCWS passwords [15].

Figure 3.1 shows the SCWS architecture.

3.2.2 SCWS Communication

The OMA has specified that the SCWS must support TLS communication i.e.

- **symmetric encryption** using (RFC4279) Pre-shared Key ciphersuites for Transport Layer Security (PSK-TLS) [231], where the communication between the SCWS and any outside entity will be protected through the use of a pre-shared key, using standardised symmetric algorithms such as AES [232].
- **asymmetric encryption** through the use of public key cryptography and certificates. The specification requires the use of the RSA algorithm [210].

The SCWS communicates with an HTTP/HTTPs client (browser) [233] based on the phone, separate from the normal SIM-to-Handset communication. There are two possible ways for this communication to take place, via the Bearer Independent Protocol (BIP) or HTTP/HTTPs:

- **BIP** is used when the SIM card does not have a TCP/IP stack. BIP is defined in ETSI TS 102 223 [234] and takes place over ISO7816. A BIP Gateway located on the phone translates TCP/IP messages to BIP commands (that the SIM understands) and vice versa: the BIP gateway encapsulates requests in a ISO7816 packet, passes the packet to the SCWS which can then retrieve the original request. The port numbers used for this are 3516 (for HTTP) and 4116 for (HTTPs). The IP address used is the "loopback" or localhost IP address of 127.0.0.1.
- **HTTP/HTTPs** is used when the SIM card implements a TCP/IP stack. Here, the HTTP/HTTPs client on the phone can access the card (and the SCWS) directly without the need for the gateway. SIM cards based on JavaCard v.3.0 Connected Edition [235] can provide this functionality. HTTP communication will use port 80, and HTTPs will use port 443: the IP address can be dynamically allocated, but the card must be addressed by the name "localuicc".

For the intra-phone communication, the SCWS acts as a server since it replies to requests by client(s) on the phone. The SCWS can also operate in client mode, and this occurs whenever the MNO or a remote trusted entity wants to update the SCWS with new content, change settings or delete/retrieve data from the SCWS.

Figure 3.2 shows client/server modes of operation of the SCWS.

3.2.3 SCWS Administration Protocols

SCWS content should be capable of being remotely updated in a secure and managed manner, similar to traditional web servers. SCWS administration protocols provide the ability to upload new data (e.g. xHTML pages), delete data and change configuration parameters for the SCWS. Depending on the amount of data that needs to be transferred, or the reason for the communication, this remote administration for the SCWS can be done using one of two standardised protocols: the Lightweight Administration Protocol (LAP), or the Full Administration Protocol (FAP):

- **Lightweight Administration Protocol**. This is appropriate when there are settings to be changed or the content of the update is small in size. The procedure

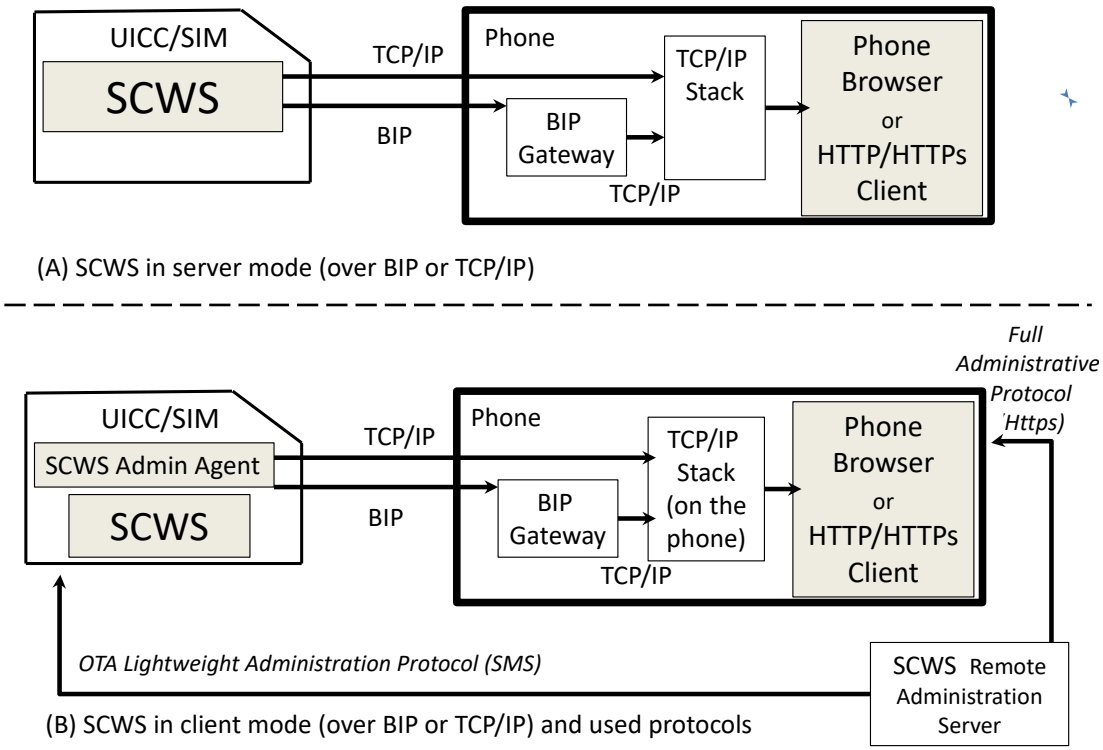

Figure 3.2: SCWS Modes of Operation

is done Over-The-Air (OTA) as described in [15]. An example would be when static HTML pages need to be transferred to the SCWS.

- **Full Administration Protocol**. This is employed when the amount of data to be sent is large and cannot fit into a small number of SMS messages. In this case the Remote Administration Server (RAS), which is defined as a trusted entity in the OMA specification, uses a Push message to communicate with an on-card entity named the Administration Agent (AA), and establishes an HTTPs connection to open an administration session. This Push message can be sent either using a formatted SMS or using the OMA SIP Push Enabler [236]. The RAS can then access and manage the data on the SIM, with applets transferred using the procedures outlined in [237]. Such communication may be triggered by on-card events, in which case the initiator is the local agent. The OMA specification defines that this connection is triggered either by the RAS or by the SIM card itself (when a certain event occurs) [15]. The FAP is used when the amount of data is relatively large e.g. loading a Java applet. If a network

connection problem occurs during a FAP session the AA attempts reconnection according to a pre-defined retry policy: if the session is abandoned, an error SMS is sent to the RAS (see [15] for details).

Both these protocols provide end-to-end authentication, integrity and confidentiality, via an authorised administration entity [15]. PUT and DELETE HTTP commands are not allowed for any entity without administration privileges, and pages that only the RAS can access must be protected using a protection set mechanism defined in the OMA specification e.g. the "/SCWS/admin" URL.

3.2.4 Access Control Policy

The OMA specification defines a security feature to protect the SCWS from unauthorised access from applications that run on the phone, the Access Control Policy Enforcer (ACPE). This determines which phone applications can be permitted to access the SCWS, thus providing a level of protection from potentially malicious applications on the mobile device. The ACPE will allow only certain trusted applications to access the SCWS, using an Access Control Policy (ACP) that will be provided to the ACPE by the SCWS over HTTP. Access to the SCWS is granted or denied based on the ACP i.e. only applications signed by the MNO, the handset manufacturer, or other trusted entities. The OMA states

> *"The ACP Enforcer is especially useful in devices that allow the user to download and install applications in the device itself (e.g. open OS phones). One use case is the download and installation of a malicious application in the handset that will try to block the access to the SCWS or ask the user for his[/her] passwords in order to access private information in the SCWS." [15]*

3.2.5 The SCWS Remote Management Ecosystem

As the MNO owns and operates the SIM/SCWS, service providers must have a business relationship with the MNO in order to install their applications on the SIM. The use of a Trusted Services Manager (TSM) to manage the business ecosystem has been suggested in the context of supporting NFC applications on mobile phones [238]. Three business models are identified: simple mode, where only the MNO can manage applications on the UICC; delegated mode, where the TSM can manage applications on the UICC but needs a pre-authorisation token from the MNO; and authorised mode where the TSM manages a specific area of the UICC without reference to the MNO. (For examples

of the key management of these business models, please see [238].) In the SCWS scenario, as the RAS is a trusted entity, the TSM could control it on behalf of the service provider.

3.2.6 Interoperability

The SCWS should provide interoperability across phone handsets and operating systems, and can be used to access web content offline using standard phone browsers. Development issues that may hinder full interoperability have been identified by the SIMAlliance [239].

3.3 Chapter Summary

This chapter described the technologies which will play an important part in the research presented later in this thesis. A brief description of the features of smart cards for telecommunication i.e the UICC (hardware) and applications i.e. SIM/USIM (software) was provided, with relevant standards. A summary of SIM Toolkit functions was also given, followed by details of the functionality and tightly controlled management procedures of the Smart Card Web Server, which will be used in proposals made later in this thesis.

Part II

Application Areas and Use Cases

Chapter 4

Remote E-Voting

Contents

Remote (Internet) e-voting uses the voter's own equipment to cast votes, but is potentially vulnerable to many common attacks, which affect the election's integrity. Security can be improved by distributing vote processing over many web servers installed in tamper-resistant, secure environments, using the SCWS on a mobile phone's SIM. A generic SCWS voting model is proposed, using a SIM/SCWS voting application with standardised MNO management procedures to process the votes cast. E-voting systems Prêt à Voter and Estonian I-voting are presented as use-cases EV-1 *and* EV-2 *which employ the generic SCWS voting model to enhance election security and protect against DDoS attacks.*

4.1 Remote e-Voting Use Cases

Remote e-voting systems have to operate in unsupervised environments, leading to opportunities for DDoS and technical attacks on the voting infrastructure, as described previously in Section 2.4.

Security can be improved by distributing vote processing over many web servers installed in tamper-resistant, secure environments, using the SCWS on a mobile phone SIM. If a SIM could be used for vote processing that would introduce a trustworthy component into the system. A mobile phone SIM is a restricted processing platform, so a voting application cannot necessarily perform all required e-voting system functions. It can provide a "front-end" input method to more sophisticated cryptographic e-voting systems which do have the required resources. With this in mind, using the SIM in e-voting will render attacks to remote e-voting system less attractive in two ways. Firstly, installing vote processing in a trusted tamper-resistant environment that can only be accessed by authorised parties will reduce the opportunity for malicious modifications to the voting application. Secondly, distributing vote processing over a large number of web servers will mean that an attacker must target multiple sites to be successful. The SCWS introduces web server functionality to the SIM environment, so a distributed vote processing application can be installed and run in a tamper-resistant environment. The use of the SIM means that the vote processing application is not accessible to an adversary who attacks the mobile phone platform.

In this chapter, a generic voting model is proposed, using a SIM/SCWS voting application with standardised MNO management procedures to transport the votes cast, that enhances election security, combats the secure platform problem and protects against DDoS attacks. The generic model is then used with two e-Voting schemes, Prêt à Voter (PAV) and Estonian I-voting. These particular schemes have been chosen as they both have features which can be readily adapted to work on a phone. Also, as the two voting schemes have different characteristics (for example, I-voting is designed for use with the Estonian national PKI infrastructure [240]) they will be presented as separate use cases, *EV-1: SCWS-PAV* and *EV-2: SCWS-I-Voting* to illustrate the flexibility of the SCWS generic model. The security requirements that need to be met by e-voting schemes are outlined in the next section.

4.2 e-Voting Security Requirements

The security requirements of a general e-voting system are to ensure that votes are: cast as the voter intended; recorded as cast; counted as recorded; and not linkable to a

Table 4.1: eVoting: Security Requirements

	Confidentiality
EV-SR1	Secrecy of the vote (*privacy*)
EV-SR2	Vote cannot be traced back to a voter (*unlinkability*)
EV-SR3	Voter can vote without external influence (*vote-buying/ coercion*)
	Integrity
EV-SR4	Votes should not be tampered with (*recorded as cast*)
EV-SR5	Votes should be included correctly in the final election result (*counted as recorded*)
	Authentication
EV-SR6	Only eligible voters can vote (*democracy*)
EV-SR7	Voters can vote only once (*democracy*)
	Availability
EV-SR8	Voters must not be prevented from voting *(forced abstention/ denial of service*)

specific voter. Also, no one should be able to determine how a voter voted. An attacker could seek to undermine these requirements and may have many goals, including manipulating the votes randomly, denying access to the voting procedure for legitimate voters, adding votes for a specific candidate/party (ballot stuffing), spoiling votes for a particular candidate or gaining knowledge about a voter's choice of candidate. By its nature, remote voting is vulnerable to coercion and vote buying, where a voter votes (willingly or unwillingly) as instructed by a third party[1]. Security requirements are summarised in Table 4.7.

The generic SCWS e-voting proposal designed to address these security requirements is now described.

4.3 SCWS e-Voting Generic Model

The proposed generic model has the following stages: registration of voters; installation of voting application and credentials onto the SCWS; voter authentication; ballot display and choosing a candidate; vote storage and sending; and confirmation that the vote has been received/ processed by the Voting Authority.

The entities involved in the proposed generic e-voting system are shown in Table 4.2; necessary assumptions are shown in Table 4.3.

[1]The proposal in this chapter acknowledges the problems caused by coercion and vote buying in remote e-voting, but does not attempt to offer a solution to these aspects.

Table 4.2: SCWS e-Voting Generic Model - Entities

Entity	**Description**
MNO	Mobile Network Operator: the MNO provides management services such as Lightweight and Full Administration Protocols to update the SCWS on the SIM, and a mobile network infrastructure.
Phone	Mobile Phone and Browser: The mobile phone handset and browser application are generally untrusted, as jail-broken operating systems or malware can compromise the correct operation of the phone's applications and operating system.
RAS	Remote Administrative Server: this was described in Chapter 3, and here provides the interface between the e-voting system and the voter's mobile SCWS. It updates the SCWS of a voter's registered phone via the MNO's FAP process, using HTTPs. The RAS can be operated by the MNO or a TTP (e.g. the Voting Authority). As explained in Chapter 3, the RAS is defined as a trusted entity in the OMA SCWS specification [15].
SCWS	Smart Card Web Server: the user accesses the SCWS environment using a PIN. The SCWS uses a Java applet for processing information securely.
V	Voter: an individual in possession of a mobile device with a SIM with SCWS installed, who uses the phone browser to communicates over HTTPs with the SCWS environment
VA	Voting Authority: The VA has details of all voter credentials, candidate information, ballot forms and election parameters. It is responsible for creating the voting application, registering voters, and receiving and counting the votes once they have been cast.
VoteAPP	Voting Application: installed on the SCWS (using installation procedures described in [237]), this displays ballot forms and collects votes ready for transfer to the VA.

4.3.1 Registration

To register, the voter must supply a mobile phone number to the VA. This will allow security credentials to be installed on the SIM/SCWS to use with the SCWS for e-voting. These are: voter ID (ID_V), voter password/PIN, voter cryptographic key pair(s) for encryption and signing (PUB_V,PK_V) and (S_V,Ver_V), and the public key of the VA (PUB_{VA}) to encrypt data sent to the VA by the SCWS (e.g. the vote). There are various ways that Voter credentials (ID/password/PIN/keys) can be obtained: for example,

- securely created by the VA at the time of registration: if the VA is responsible for generating voter credentials, recommended best practices should be followed e.g. NIST SP800-57 Part1 [241]. Additionally the public key of the VA (PUB_{VA})

Table 4.3: SCWS e-Voting Generic Model - Assumptions

	Description
EV-A1	The MNO has authorised the VA to use a RAS to update SCWS applications and data: the VA may need to have trusted business relationships with several MNOs to maximise the availability of the voting application.
EV-A2	The SCWS has a one-to-one mapping to a user, i.e. only one registered voter can use a particular SCWS.
EV-A3	A secure registration procedure is in place: the voter will supply a mobile phone number to the VA and authorise its use so that the RAS will be able to download the Java applet/credentials onto the correct phone using techniques described in Chapter 3. The voter will also set a PIN/ password to access the SCWS environment.
EV-A4	The HTTPs channels between the RAS and SCWS, between the SCWS and Browser, and between the RAS and VA are considered secure.
EV-A5	The VA is trusted not to collude with the MNO: the MNO is trusted not to collude with any other entity, especially other network operators (i.e. there should be a "circle of trust".) The MNO only provides management procedures and infrastructure and should not be trusted with sensitive voter, ballot form or election information.

Table 4.4: SCWS e-Voting Generic Model - Protocol Notation

Notation	**Description**
E_K(Z)	Encryption of data Z with key K
ID_X	Identity of entity X
MNO	Mobile Network Operator (entity)
N_X	Random Nonce generated by entity X
PK_X/ SK_X	Public/ Secret Key pair of entity X. The key size should be according to best practices (see NIST SP800-57 Part1 [241] for details). Key pairs could be obtained from a national digital ID scheme if available.
RAS	Remote Administration Server (entity)
S_X/ Ver_X	Signing/ Verification key pair of entity X. The key size should be according to best practices (see NIST SP800-57 Part1 [241] for details). Key pairs could be obtained from a national digital ID scheme if available.
V	Voter (entity)
VA	Voting Authority (entity)
X→Y:	Message sent from entity X to entity Y
(Z)$Sign_K$	Signature on data Z with signature key K

Table 4.5: SCWS e-Voting Generic Model - Security Credentials

Credential	**Description**	**Location**
ID_V	Voter ID: Size and complexity according to VA's policies - it is better to avoid having a voter ID similar to the voter's surname or other demographic information, so that it will be more difficult for an attacker to guess.	Stored on the SIM. Known by voter and VA.
Password	Size and complexity according to best practices: e.g. at least 8 characters long with upper/lowercase letters, at least a number and a punctuation character [242].	Stored on the SIM. Known by voter and VA.
(PUB_V,PK_V) (S_V,Ver_V)	Cryptographic key pairs (public/private) (PUB_V,PK_V) used for encryption/decryption, (S_V,Ver_V) for signing/verifying - for key separation purposes, it is better to have different key pairs for these cryptographic functions (but this depends on the VA). Key sizes should follow best practices (see NIST SP800-57 Part1 [241] for details). Key pairs could be obtained from a national digital ID scheme if available.	Private keys PK_V, S_V stored on the SIM alone. Public keys PUB_V, Ver_V also known to the VA.
PUB_{VA}	The Public Key of the VA will be used to encrypt data sent to VA by the SCWS (e.g. the vote). It is best practice to use a different key for every election, so that brute force attacks against it (if successful) only affect one election .	Stored on the SIM and known to everybody

should be different for every election, so that brute force attacks against it (if successful) only affect one election.

- obtained from a national digital ID scheme if available. If a government-issued identity scheme is used to supply voter credentials, there will be a very strong independent link between the SCWS and voter identity, which will make it very difficult for a voter to be impersonated. NFC phones and contactless smartcard ID cards could provide an alternative, easy to use method for authenticating voters, using the NFC phone as a smartcard reader. This requires an application on the untrusted mobile handset, but authentication credentials can be protected in transit through the phone by encryption. NFC can also allow the use of an identity token with a strong independent linkage to identity, such as a passport or government-issued identity card, e.g. as in [243].

- key pairs could also be generated by the SIM and sent to the VA.

4.3.2 Installation of Application onto SCWS

Credentials are sent to the voter's SCWS by the RAS, using FAP. Firstly, (PUB_V,PK_V), (S_V,Ver_V) and PUB_{VA} are generated and sent to the SCWS[2]. After this procedure has completed successfully, the ID_V/Password and application (code/data/optional ballot forms) are transferred in the same way, encrypted by PUB_V for decryption on the SIM using PK_V. The communication channel in both occasions is protected by HTTPs. This is shown as Messages 1 and 2 in the protocol diagram in Figure 4.1.

The voting application (a Java applet running on the SCWS) must be given access to the voter credentials and the keys installed on the SIM, and is able to create dynamic content [15]. The application creates this content whenever requested and returns it back to the SCWS (which in turns serves this content to the voter as HTML pages).

The voter may be notified about the installation, for example via an SMS, an automatic call (such as an Interactive Voice Response (IVR)), an e-mail to a pre-registered e-mail address or by simply updating the SCWS home page with a link pointing to the voting site inside the SIM card. This is shown as Message 3 in Figure 4.1.

4.3.3 Authentication

The VA will again contact the voter through an SMS, IVR call or e-mail when the election starts, and the SCWS homepage will show a "VOTE NOW" link . Clicking the link transfers the voter to the SCWS environment and an authentication page is displayed. In this page the voter will enter the ID_V/Password/PIN issued during the electoral registration stage. The ID_V/ Password/PIN are checked against the ones that were previously transferred to the SIM card, and that the ID_V has not been used to vote already. If the authentication is successful, the voter is presented with the ballot form, in a format determined by the e-voting system used. (Optional Messages 7a/7b can be used for additional processing at the VA: e.g. if the ballot form was not installed on the SCWS by Message 2, it can be retrieved using the ID_V to obtain the appropriate list of candidates; or voter eligibility can be checked against centrally held databases). Again all the communication takes place over HTTPs, and this stage is shown as Messages 4 to 7 in Figure 4.1.

4.3.4 Choosing a Candidate

The ballot form allows the voter to choose a candidate, with a text entry option if required by the voting scheme used. Selecting a "VOTE" button generates the confir-

[2]If the keys are generated in the SIM environment and sent to the VA, this will result in an extra message in the protocol: for simplicity, the protocol illustrates the VA installing keys on the SCWS.

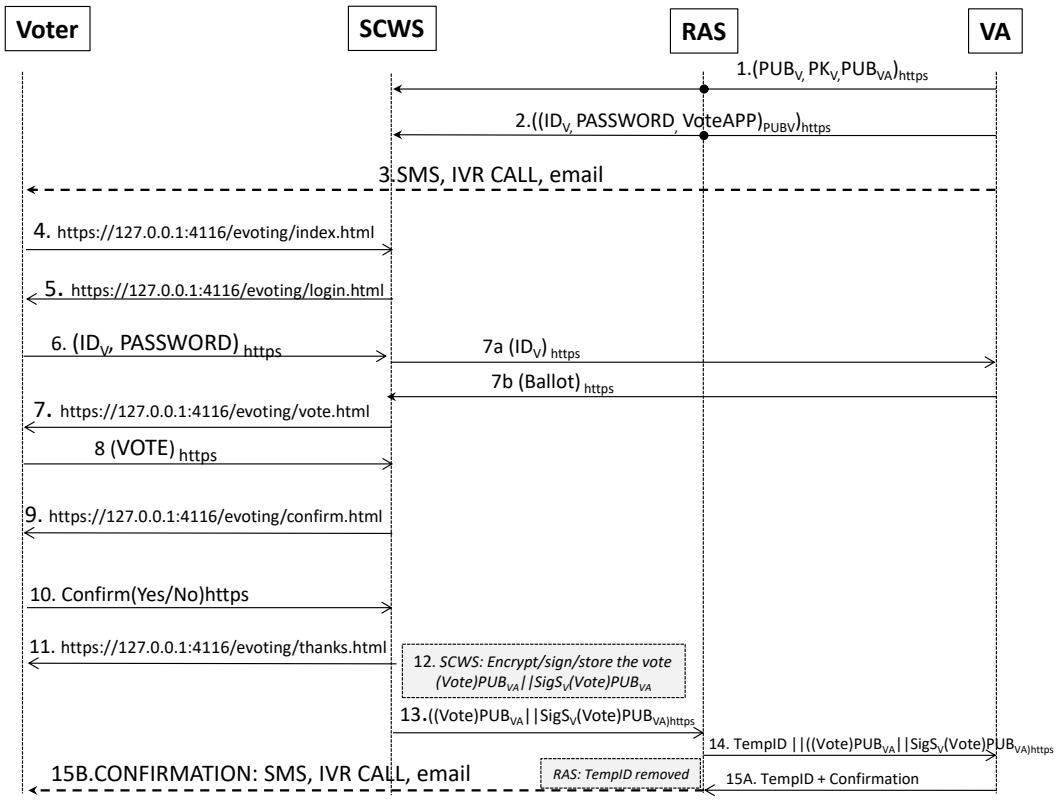

Figure 4.1: SCWS Voting Generic Model - Protocol

mation screen, where a simple yes/no, or a PIN associated with the ID_V can be used to confirm the voter's choice. Once the voter confirms, the vote is submitted to the SCWS. These are Messages 8 to 11 in the protocol diagram in Figure 4.1 and again all communication is over HTTPs. Alternatively, it is possible to select a dummy vote process rather than a real vote, if the e-voting system allows this for verification purposes e.g. as seen in the proposal by Chaum et al. [244]. This is intended to allow the voter to challenge the e-voting system and gain reassurance that the cryptographic processing is working correctly. Once the voter confirms, the vote is submitted to the SCWS.

4.3.5 Vote storage and sending

The vote is now stored on the SIM card. The format of the vote is dependent on the voting scheme used: for example, it could be a pre-assigned vote code as in [18], or a cryptogram which determines the position of the candidates on the ballot form as in [81]. The vote is encrypted with PUB_{VA} (and signed if required), and this will be

retrieved by the VA and subsequently deleted from the SCWS after it is cast. The vote is also kept on the SIM for future reference, encrypted with PUB_V. This encryption is not strictly necessary, but is included to provide an additional security measure in case space restrictions mean the vote is stored in non-tamper-resistant memory or on the phone. A flag is set to indicate that the ID_V has been used to vote. Once the vote is encrypted, the SCWS administration agent will either trigger a connection with the RAS, or the RAS can automatically retrieve the vote on a specific day/time. In both cases the two entities initiate an HTTPs connection and the vote is collected via the FAP. The RAS passes the signed vote (unchanged) to the VA, tagged with a temporary ID so that the vote receiving process at the VA does not find out the mobile number, as this could be used to link the voter to a particular vote. Once the vote is received by the VA, the voter receives a notification that their vote was cast via a second channel e.g. an SMS message, IVR or an email. Steps 12 to 15a/b in Figure 4.1 show the vote storage and sending process. Using the SCWS model keeps all sensitive vote information private in a tamper resistant environment, so that no one can determine how a voter voted.

4.3.6 SCWS e-Voting Generic Model: Summary

The SCWS e-Voting generic model presented here inherits many desirable security properties from its use of a) the tamper-resistant environment of the SIM for vote storage and processing, and b) the standardised and tightly controlled management procedures associated with the SCWS. A security analysis is shown later in this chapter, in Section 4.6.

The next section describes the first e-voting use-case, *EV-1: SCWS-PAV* which demonstrates how the generic SCWS model can be used with the Prêt à Voter e-voting system.

4.4 EV-1: SCWS-PAV

4.4.1 Prêt à Voter - Background Information

Prêt à Voter (PAV) [81, 244, 245, 246] is a paper-based e-voting system designed for the supervised environment of an election polling-station. The paper ballot form is used as input to a cryptographic system. The electoral process can be audited by voters and third parties, to give end-to-end verifiability. Voters can verify the system in two ways: by performing dummy votes not included in the final tally, which are intended to check that the cryptography used in the ballot form is correct; and by checking that

their vote appears on a secure Web Bulletin Board (WBB) once it has been cast.

Voters are given a paper ballot form, and they mark their choice with an X. Every ballot form is different, as candidates are listed in a (different) random order on each one. There is a code (called an "onion") which contains details of the candidate order in encrypted form. When the vote is cast, the left hand side of the form (containing the candidate names) is detached and destroyed, leaving the voter with a voting slip showing the position of their vote on the form, but not who the chosen candidate was. The vote is input to the PAV system and the voter is given the slip as a receipt which can be checked against a web bulletin board at a later stage. The actual vote recorded in the system is the numerical position of the voter's choice and the onion i.e. (*index, onion*).

The PAV scheme was extended in [246] by including confirmation codes for each candidate, printed on the ballot form. These codes are calculated by the VA prior to the election. Once a voter has cast their vote, the confirmation codes are recalculated by the VA and relayed back to the voter at the poll-site. The voter can check this received value against a confirmation code hidden on the ballot form under a scratch-off strip, which only they should see. The voter's receipt contains the position (index) of their chosen candidate, the associated confirmation code and the onion. Figure 4.2 shows a paper ballot form with confirmation codes, before and after voting, with the onion in the bottom right corner.

In the PAV scheme, ballot forms can be printed on demand by the voting booth, as described in [245]. Briefly, two related onions are calculated for each ballot form, the "booth onion"(left onion) encrypted with the public key of the voting booth, and the "registrar onion" (right onion) encrypted in the normal PAV way. These "proto-ballot" forms can be distributed and stored securely prior to the election. Alternatively, the left onion could be encrypted with PUB_V (rather than a booth key) without losing any security, by using ElGamal for distributed blinding as described in [247]. On Election Day, when a voter takes a ballot form into the voting booth, the booth reads the left onion, uses its previously stored secret key to decrypt the onion, reconstructs the candidate list and prints the ballot form. The vote is cast using the right onion, i.e. (*index, right onion*).

4.4.2 Using the SCWS with Prêt à Voter

It is the ability to print ballot forms on demand, along with the use of confirmation codes that make PAV a suitable example e-voting system to use with the SCWS model. The advantage of using PAV with confirmation codes is that most of the cryptography (i.e. generating the onions) is done by the VA before the election starts, so this min-

Name	Vote	Code
Bob		
Charlie		
Alice		
		127645

Vote	Code
X	3672
	127645

Figure 4.2: Prêt à Voter Ballot Forms Before and After Voting

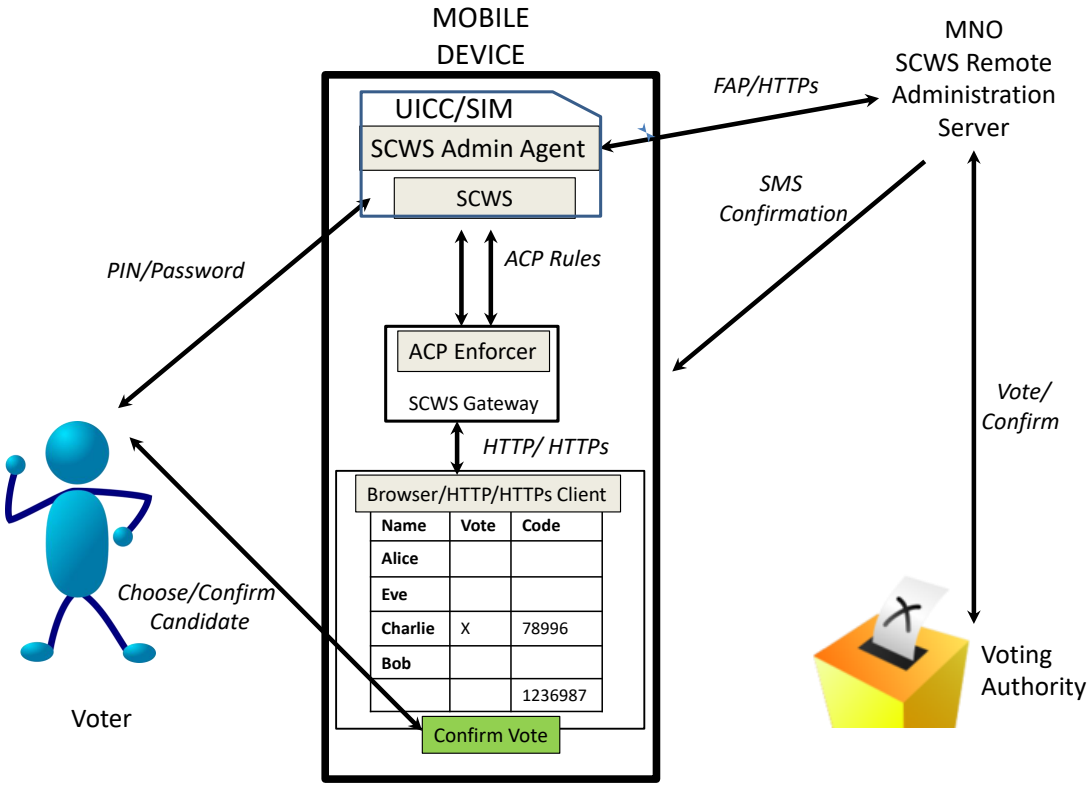

Figure 4.3: EV-1: Prêt à Voter and SCWS

imises the amount of cryptographic processing necessary in the restricted environment of the SIM. With suitable modifications to the generic model, the SCWS voting application can play the part of the PAV voting booth, to decrypt the candidate ordering and hence display ballot forms. The SCWS can also store confirmation codes securely so that when the voter receives a code from the VA (in an SMS) the two values can be compared. Figure 4.3 shows an overview of the design.

Installation of Keys and Application onto SCWS

In Message 2 of the protocol shown in Figure 4.1, PAV candidate lists and proto-ballots (including confirmation codes) are sent to the SCWS along with the voting application and other credentials. The left onion must be encrypted with PUB_V, using the ElGamal blinding technique mentioned earlier. It is suggested that several proto-ballots are sent, as this allows the voter to cast dummy votes to audit the system if they so desire.

Authentication

The authentication stage remains the same as the generic SCWS approach, but when the ballot form is to be displayed (Message 7 in Figure 4.1), the PAV method of constructing the candidate order must be followed. This involves decrypting the left onion using ElGamal and PK_V, and then reconstructing the candidate order using a predetermined PAV procedure [244, 245]. The ballot form can then be displayed (minus confirmation codes).

Choosing a Candidate/Vote Storage

Choosing the candidate and voting is the same as in the generic SCWS approach. The confirmation code for the selected candidate is displayed along with Message 11. Once the candidate has been chosen, the vote needs to be signed with the voter's signing key before it is retrieved by the RAS. A copy of the vote with its associated confirmation code is encrypted with PUB_V and stored on the SCWS. Storing the confirmation code on the SCWS ensures that at a later stage, only the voter can see this code (as specified in PAV). Finally, a flag will be set on the SCWS to indicate that the voter has voted.

Vote Sending

The RAS passes the signed vote (unchanged) to the VA, as in the generic model. Once the VA has received the vote, checked its validity, and posted it to the bulletin board, it will recalculate the confirmation code for the chosen candidate, and send it and the temporary ID to the RAS. The confirmation code can then be returned to the voter via a second channel (SMS/IVR/email), for checking against the one previously displayed on the phone browser and stored on the SCWS. This provides assurance that the vote was counted as cast because the correct confirmation code can only be generated at the VA once it has been successfully decrypted and matched to a valid entry in the bulletin board.

Dummy Vote Procedure

A dummy vote should check that the confirmation codes associated with all the candidates shown on a ballot form are correct, to provide assurance to the voter that the underlying cryptographic calculations are accurate. Choosing the dummy vote option will cause the SCWS to create a different message for the VA (via the RAS) containing *(audit request, right onion)*, rather than the usual vote. The VA will recalculate the confirmation codes for all the candidates on the ballot form, and send them to the voter via the second channel: the dummy vote will not be counted. The SCWS will display all the stored candidate confirmation codes via the browser, for the voter to check against those received from the VA. The ballot form on the SCWS which was used for the dummy vote would then be marked as not selectable for future voting. If all the stored ballot forms have been used for dummy votes, optional Message 7a and 7b shown in Figure 4.1 can be invoked to obtain another batch of ballot forms from the VA.

4.4.3 Use Case EV-1: Summary

The verification procedures of PAV gives the voter the assurance that their vote has been cast as intended, and counted as cast. Using the SCWS model keeps all sensitive vote information private in a tamper resistant environment, so that no one can determine how a voter voted.

The next section now presents the second e-Voting use-case, *EV-2: SCWS I-Voting*, which demonstrates how the SCWS e-voting generic model can be used with the Estonian I-Voting scheme.

4.5 EV-2: SCWS-I-Voting

4.5.1 Estonian I-Voting System - Background Information

Internet voting has been used for elections in Estonia since 2005, underpinned by the strong authentication provided by ID cards which use the Estonian National Public Key Infrastructure (PKI) system for identification and digital signatures [240]. Known as I-voting, the Estonian scheme employs the "double envelope" system, similar to postal voting, where the vote is encrypted with the VA public key (the secret "inner envelope"), and digitally signed by the voter (the "outer envelope"). Internet voting is possible during a 7-day period from the 10th to the 4th day prior to election day itself, and voters are allowed to change their vote during this time by re-voting electronically or visiting a polling station and casting a paper vote. Only the last recorded vote

will count: paper-based votes cancel out electronic votes. The facility to re-vote is intended to provide some protection against coercion and vote-buying: duplicate votes are manually cancelled by election officials. A mock "test" election is run before a real election to identify potential problems e.g. whether voter equipment has the correct settings or not. Internet voting is popular: 24.3% of participating voters in the 2011 Parliamentary elections used I-voting, and there is "widespread trust in the conduct of the Internet voting" [95]. For full information about the Estonian I-voting system please see [87].

Please note: the descriptions that follow in use case *EV-2* are based on the Estonian I-Voting Scheme as it operated until 2015: there had been criticism from the academic community and others [248, 249], and the scheme has since been updated to include more robust security features and provide voter-verifiability through the use of a QR-code confirmation of the vote cast [250, 251]. The new architecture [19] is summarised in Appendix A.2.1.

Estonian ID Cards and Mobile-ID

Estonian ID cards used in conjunction with the National PKI scheme have two PINs, for use in authentication and signing: they are commonly employed to access a range of online government and financial services. Since 2011, there has been an alternative to using a physical ID card for online authentication, called Mobile-ID. This uses a specially issued SIM, obtained from an MNO, which holds public/private key pairs and certificates that are activated by the Police and Border Control Department. The security of the Mobile-ID protocol has been investigated by [252], and the protocol was found to be at least as secure as authentication with a physical ID card. Using Mobile-ID, a phone can be used alongside a PC for authentication instead of a smart card reader. A SIM Toolkit application [29] performs the necessary encryption/ signing. There is a very strong link between voter identity and the Mobile-ID SIM.

I-Voting

To use I-voting, the voter must download a voting application onto their PC, and identify themselves either using an ID card inserted into a smart card reader attached to the PC or by Mobile-ID. Using an ID card means all the voting stages are done via PC, firstly entering PIN1 to identify the voter and retrieve the ballot form, and secondly entering PIN2 to sign and cast the vote once the choice is made. The Mobile-ID procedure is more complex, and involves an external, trusted Certificate Authority (CA) to verify signatures/certificates. A voter identifies themselves by entering their

mobile number into the downloaded voting application. This results in identical control codes being sent from the CA to the PC and to the mobile (via SMS). The voter must input their identification credential PIN1 into the mobile to confirm that the control codes received are identical. The ballot is then displayed on the PC, and the voter makes their choice on the PC. A new pair of identical control codes is sent from the CA to the PC and phone, to be checked by the voter (that they are the same). If so, the voter confirms their vote by inputting their digital signature credential PIN2 into the phone. The vote is signed and cast, and a confirmation message then appears on the PC.

I-Voting System Architecture

The I-voting system contains several components, shown in Figure 4.4A: these include a publicly-accessible Vote Forwarding Server (VFS), which interacts with a Vote Storage Server (VSS) and Voter/Candidate databases behind the VA's firewall. Each voter has a Personal Identification Code (PIC) on the Voter database: this database is updated daily during the voting period, with amendments to voter information. An offline Vote Counting Application (VCA) tallies the votes cast at the end of the election. The vote itself is formed as follows: *(candidate choice, random number)*, encrypted with the public key of the VCA, signed with the voter's signing key S_V, and the voter's certificate $Cert_V$. The VFS checks whether the individual who authenticated themselves at the start of the session is the same one who gave the digital signature, and then forwards the vote to the VSS. The VSS checks the correctness of the digital signature via an external CA (also called the validity confirmation server). The VSS acquires a certificate $Cert_{VALID}$ that confirms the validity of the digital signature which is then added to the signed vote.

4.5.2 Using the SCWS with Estonian I-voting

It is the use of Mobile-ID in I-voting that is of interest here: the Mobile-ID SIM has a very strong link to a person's identity, and has cryptographic key pairs installed for public key encryption/signing. If a Mobile-ID SIM also has SCWS capabilities, the SCWS generic voting model could be used with the Estonian I-Voting system to provide a secure, fully mobile voting interface, without the need for a PC and smart card reader/ phone. An illustration of the proposed SCWS voting alternative is shown in Figure 4.4B.

It can be seen in Figure 4.4 that the voter's PC plus smart card reader/mobile phone is replaced by a phone with the SCWS voting application. The SCWS voting

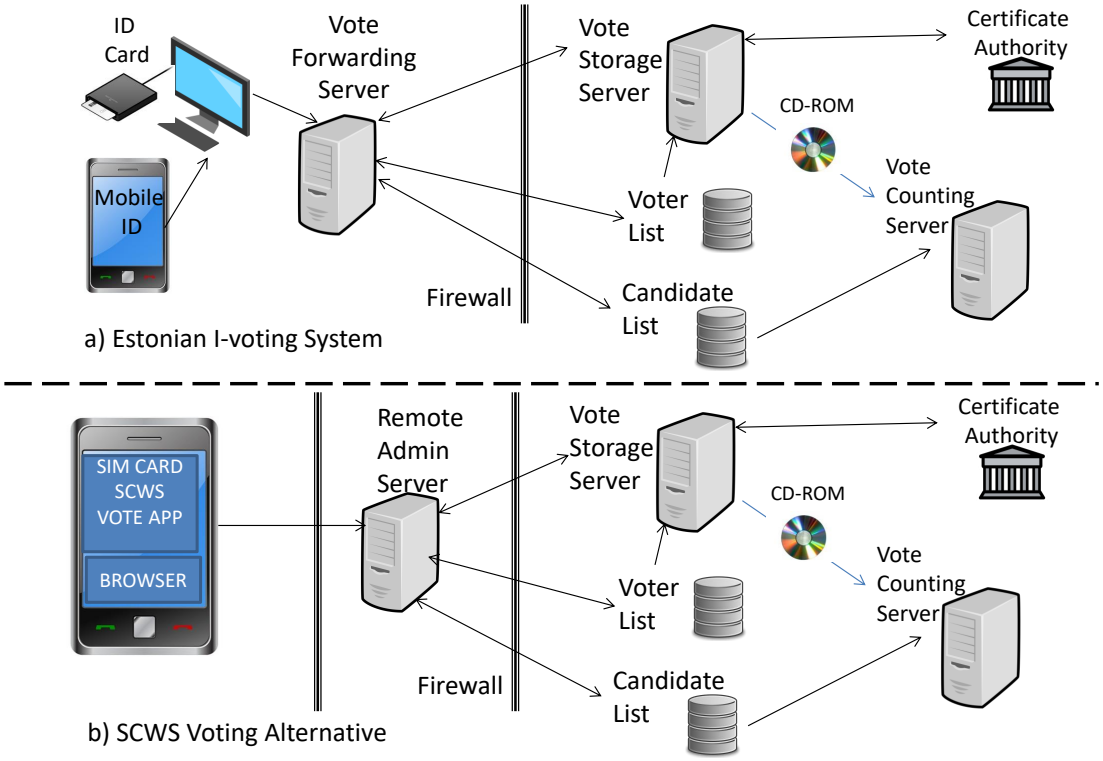

Figure 4.4: EV-2: Estonian I-Voting System and SCWS Architecture

application also performs the role of the VFS (e.g. checking the voter's authentication credentials). This means that there is no publicly accessible server in the SCWS solution, only the trusted MNO-controlled RAS. Control codes from the CA to the mobile are not necessary during the SCWS voting process, as all the relevant keys and PINs are stored in the SIM and accessible to the SCWS voting application.

The VCA and PIC equate to the"VA"and "ID_V" in the previously described SCWS generic model respectively: in the following description, the terms "VA"and "ID_V" will be used for consistency.

The SCWS I-voting procedure is shown in Figure 4.5, and is now outlined.

Installation of Keys and Application onto SCWS

In the existing Estonian system, a voting application is downloaded to the voter's PC: this stage needs to be modified to include an option to choose SCWS voting. Here, the voter must authenticate themselves to the VA using the standard Mobile-ID procedure outlined previously (Message 1 in Figure 4.5). This is to ensure that the

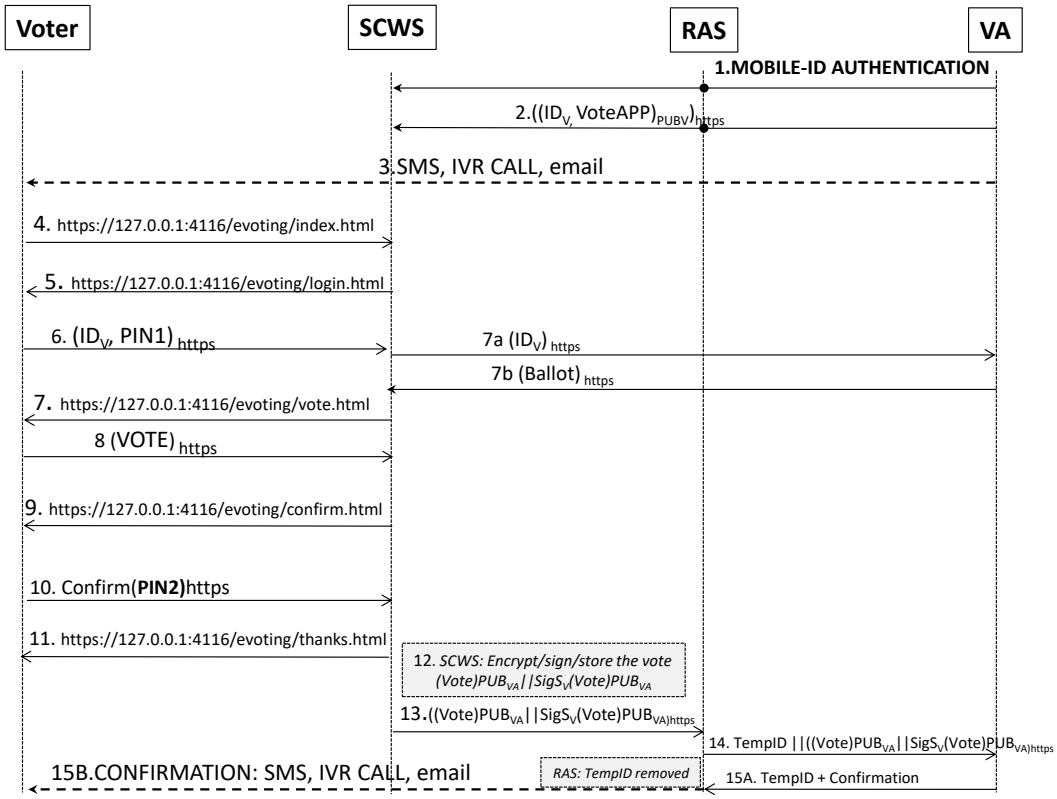

Figure 4.5: EV-2: SCWS I-Voting - Protocol

voter is in possession of the Mobile-ID SIM that the SCWS voting application will be installed on, and that the VA knows the public key of the voter (PUB_V) with which to encrypt the application installation files. As Mobile-ID SIMs already have key pairs for encrypting/signing along with corresponding digital certificates and PINs (1 and 2), there is no need to install any further voter keys/passwords. Message 2 of the protocol shown in Figure 4.5 can now be used to install the voting application and public key of the VCA (PUB_{VA}) onto the SCWS. Ballot forms could be installed on the SCWS at this stage if desired.

Authentication

Once the advance voting period has commenced, the authentication stage in the generic SCWS model (Messages 4-6) can use PIN1 associated with the voter's Mobile-ID to access the voting application. After successful authentication, the ballot form can either be retrieved from the SCWS voting application (if it had been installed on the SCWS in Message 2), or optional Message 7a can be used to send the ID_V to the VA (via the

FAP/RAS) to retrieve the ballot from the VA's centrally held candidate and voter lists and check voter's eligibility. Message 7b will then return the ballot form.

Choosing a Candidate/Vote Storage

Choosing the candidate and voting is the same as in the generic SCWS approach. Once the candidate has been chosen, the vote is formed by using PUB_{VA} to encrypt the voter's choice along with a random number. The digital signature credential PIN2 is used by the voter to confirm the vote, rather than through a simple yes/no choice. The vote will then be signed with the voter's Mobile-ID signing key S_V, once the voter has confirmed their choice. There is no need to set a flag indicating that the voter has voted, as re-voting is permissible and back-office procedures at the VA are used to remove duplicate votes. The vote itself is formed as follows: *(candidate choice, random number)*, encrypted with the public key of the VCA PUB_{VA}, signed with the voter's signing key S_V, along with the voter's Mobile-ID certificate $Cert_V$.

Vote Sending

Sending and confirming the vote is as described in the generic SCWS e-voting model i.e. the vote is retrieved by the RAS and sent to the VSS. The voter's certificate of validity $Cert_{VALID}$ is added to the vote by the VSS, after checking the correctness of the digital signature via an external CA (validity confirmation server). The confirmation message returned to the voter via SMS/IVR/email will show that the vote has been received by the VA and will be stored and duly processed once the advance voting period is over.

Dummy Vote Procedure

There is no "dummy vote" procedure as such, but there is a mock election stage, where settings etc. can be tested.

4.5.3 Use Case EV-2: Summary

The (pre-2015) Estonian I-Voting system is not voter-verifiable, so does not give the voter the assurance that their vote has been cast as intended, and counted as cast. The advantages of using the SCWS model is that it keeps all sensitive vote information and processing private in a tamper resistant environment, so that no one can determine how a voter voted, and there is no centralised voting server that can be attacked in a DDoS exploit.

A comparison of the SCWS generic model with use cases *EV-1* and *EV-2* is shown in Table 4.6.

Table 4.6: Comparing Generic Model to EV-1 and EV-2

Stages	**Generic Model**	**PAV: EV-1**	**I-VOTING: EV-2**
Register	Generate PUB_V, PK_V, PUB_{VA}	Generate PUB_V, PK_V, PUB_{VA}	MOBILE-ID
Install	App/keys sent via FAP	+"Proto Ballots"	PUB_{VA} sent via FAP: MOBILE-ID already installed
Authenticate	ID_V, PIN/Password to SCWS	as per generic model	MOBILE-ID PIN1
Display Ballot	Optional Msg 7a/b Display webpage	Decrypt LO: PAV method to extract ballot	as per generic model
Choose **Candidate**	Select candidate Confirm Y/N	Vote=(index, RO) Confirm Y/N	Vote=(choice, Nonce) Confirm with PIN2
Dummy Vote	Select "audit", vote discarded. Msg 7a/b for more ballots	Receive All Confirmation Codes Ballot discarded	No Dummy Vote
Store Vote	(Vote)PUB_{VA} Signed S_V, Voting Flag set, (Vote)PUB_V Stored	as per generic model	No Voting Flag set: Re-voting is allowable
Send Vote	(Vote)PUB_V Signed S_V, RAS retrieves via FAP	as per generic model	as per generic model
Confirm	Confirmation sent via second channel (SMS/IVR/email)	Confirmation code shows vote is received and counted	Confirmation shows vote is received

The next section now provides a preliminary analysis of the security of the two use cases, *EV-1* and *EV-2*.

4.6 Security Analysis

The security properties of the SCWS installed on the tamper-resistant SIM are described in more detail later in this thesis, in Chapter 8. The main advantages of using the SCWS solution for e-voting are that the SCWS, the voting site and the credentials are stored in a secure token, the SIM card, that has defences against physical and side channel attacks. Also, distributing the voting application to many SIMs means that even if an attacker manages to overcome SIM defences they will only gain access to one voter's credentials: the voting application code is equally difficult to attack. Forcing an attacker to target a large number of phones and their SCWS/SIMs means that attacks are hard to scale and need extensive efforts to be effective, giving DDoS protection. In contrast, existing remote e-voting schemes have faced common internet application problems such as DDoS attacks [97] and SQL injections [96] and technical failures [94]. Additionally, secure platform issues arise because the voter's equipment cannot be trusted to perform as expected either through malware or system vulnerabilities e.g. [95].

Use cases *EV-1* and *EV-2* are now discussed with respect to the security requirements set out in Section 4.2: potential attack goals and mitigations are identified.

4.6.1 Attack Goals

An attacker who seeks to undermine e-voting security requirements may have many goals, including manipulating the votes randomly, denying access to the voting procedure for legitimate voters, adding votes for a specific candidate/party (ballot stuffing), spoiling votes for a particular candidate or gaining knowledge about a voter's choice of candidate. By its nature, remote voting is also vulnerable to coercion and vote buying, where a voter votes (willingly or unwillingly) as instructed by a third party.

The most relevant attacks against remote e-voting security requirements are mitigated as follows:

Confidentiality (EV-SR1/EV-SR2/EV-SR3):
Attack Goal - Find out contents of vote, or how a voter voted
This attack could be done by retrieving the vote from the SCWS. The vote is stored inside the SIM card; encrypted using PUB_{VA} and is deleted once the vote is cast. So the vote is always held encrypted in the secure token and inaccessible to an attacker.

The SCWS can only be accessed from the handset's HTTP client and the RAS, thereby minimising the likelihood of attacks that occur remotely. An attacker would need to be in possession of the voter's phone and know the voter's credentials in order to access the SCWS voting site. The vote could be targeted when it is in transit between the SCWS and the VA, but this communication channel is protected by HTTPs, and the RAS is a trusted entity. The voter's credentials are securely stored in a tamper resistant token. Thus the confidentiality of the scheme can be reasonably assured.

Attack Goal - Prevent voter from voting

It must be noted that coercion (EV-SR3) is a generic problem which is not solved by the use cases presented here. It is possible to use coercion resistant e-Voting schemes with the SCWS generic model, as the system could be modified to include fake credentials and/or duplicate (chaff) votes as seen in Civitas [253] or to use panic passwords [254]. However, combining these with voter-verifiability is an open problem, as it is difficult to check if a vote has been cast as intended when there are elements of a voting system designed to discard votes received but produce a valid receipt. The Estonian I-voting system uses multiple re-voting as an anti-coercion measure. This is a compromise of security and usability, as manually discarding multiple votes introduces new attack points into the voting system.

Integrity (EV-SR4/EV-SR5):

Attack Goal - Change contents of vote

Tampering with the content of the vote could be attempted by the phone browser, or when the vote is in transit. The vote in transit is protected by the security of HTTPs for all communication channels. In *EV-1: SCWS-PAV*, any type of tampering with the vote can be detected by the use of confirmation codes and a WBB where the voter may check their vote was included as cast. This is the advantage of using a voter-verifiable scheme, which ensures security requirements EV-SR4 and EV-SR5 are met.

However, in the pre-2015 Estonian I-voting scheme, a vote could be changed by a malicious application without the voter's knowledge. This is a secure platform issue, which is not well addressed by this version of the Estonian I-voting scheme, even for PC-based Internet voting. This inherent vulnerability is carried through the the SCWS use case: the phone browser could change the vote undetected. An advantage of the SCWS voting approach is that the voting application itself cannot be easily manipulated by malware on the phone, unlike the Estonian PC voting application which is vulnerable to malware [95]. This SIM based solution provides more security in that respect. Tampering with the SCWS voting application to change the content of the vote can only be done before it is installed on the SCWS/SIM. This latter possibility would

require an insider attack at the VA or MNO, which would likely require sophisticated and well resourced planning. So weaknesses in the e-voting scheme mean that in this use case, EV-SR4 is not met. Also, as the pre-2015 Estonian I-voting scheme is not verifiable by the voter, there is no way to check that votes are cast correctly, so EV-5 is not met either.

Authentication (EV-SR6/EV-SR7):
Attack Goal - Vote more than once
In *EV-2: SCWS-I-voting*, voters are allowed to vote multiple times during the advance polling period, so in this case, voting more than once using the SCWS solution is not a security issue per se: back-office procedures identify and discard duplicate votes. However, tampering with the SCWS application could generate votes without the voter's knowledge. As mentioned previously, it should be practically infeasible to tamper with the SCWS voting application from the phone handset, and remote access is only possible through a trusted entity, so generating false votes that way should not be possible. However, if an insider attack at the VA interfered with the SCWS application such that it produced unauthorised votes for the RAS to retrieve, this would not be identified by the pre-2015 Estonian I-voting system as it is not voter-verifiable.

In voting schemes where multiple votes are not permitted as in use case *EV-1: SCWS-PAV*, an attacker might attempt the following: use one registered voter's credentials several times on one or more phones; steal multiple ID_V/Passwords/PINs and vote on one or more phones; generate fake voter IDs; or mount insider attacks at the VA. The countermeasures for each type of attack are as follows:

- If an attacker has gained access to the registered voter's ID/Password/PIN during the voting process, the credentials cannot be used to vote on the same phone more than once. The ballot form will only be displayed when a ID_V has not been used before, as the SCWS voting application sets a flag to indicate that the ID_V has voted. The VA must also have back-office procedures to make sure that an ID_V is not used to vote more than once. If a voter's credentials are used on a different SCWS to the one registered for that voter at the VA, they would not be recognised: voter credentials can only be used to access one registered SCWS.

- An attacker may obtain multiple ID_V/Passwords/PINs, perhaps by coercion, and attempt to vote many times on one phone. However, the application on the SCWS will only recognise the credentials of the voter registered to use that particular SCWS, and so multiple voter credentials cannot be used to vote on a single handset. To vote successfully on several phones, the attacker would not only

need multiple credentials, but also access to the associated physical SIM/SCWS and its voting application, which makes this a much more difficult attack to mount.

- Generation of fake ID_Vs should not be possible if stringent registration procedures have been followed.
- Insider attacks at the VA resulting in the generation of false votes will be detected by the PAV scheme because the voter would receive spurious confirmation messages from the VA.

Denial of Service (EV-SR8):
Attack Goal - Prevent voter from voting
Voters could be prevented from voting by physical means (e.g. lost/stolen equipment), by subverting the correct operation of the proposed voting solution, (e.g. DDoS attacks), or by technical measures which suppress the vote and do not allow the VA to retrieve it from the SCWS. The SCWS generic voting model is DDoS resistant: its main strength is that distributing the voting application to the SCWS makes the system resilient to centralised web server attacks, as the application is installed in many attack resistant SIMs. There will need to be procedures in place to allow poll-site voting for those whose handsets are lost/stolen/malfunctioning. In the SCWS solution it is very difficult to suppress the vote by technical means: to launch a successful attack, individual phone handsets must be targeted. In both *EV-1* and *EV-2*, an SMS confirmation message sent to the voter will give assurance that their vote has indeed been retrieved and received by the voting authority: the confirmation code in the PAV system additionally denotes that the vote will be counted correctly by the VA. If no SMS is received, then the voter can raise a query with the VA and use an alternative method to vote.

There is no centralised voting server to be attacked, hence a single point of failure has been removed. DDoS attacks cannot easily take place because the voting processing is dispersed among many voters. Each voter effectively has a voting server on their SIM, so for an attack to have a significant outcome, the attacker has to infiltrate a large number of phones and find a way to orchestrate an attack against them. A potential attacker must also find out which phones are registered for voting in order to target them.

The RAS could be considered a single point of failure, but it may have restricted functions and limited access as it runs on TTP/MNO premises: it should be isolated from any unauthorised physical access. However, although internal threats cannot be

Table 4.7: eVoting Use Cases vs Security Requirements

Security Requirement	**EV-1**	**EV-2**
EV-SR1 (Privacy)	✓	✓
EV-SR2 (Unlinkability)	✓	✓
EV-SR3 (Coercion)	×[a]	✓[b]
EV-SR4 (Recorded as cast)	✓	×[c]
EV-SR5 (Counted as recorded)	✓	×[c]
EV-SR6 (Democracy)	✓	✓
EV-SR7 (Democracy)	✓	✓
EV-SR8 (Denial of Service)	✓	✓

[a] Not a coercion-resistant e-voting scheme.
[b] Re-voting allowed as anti-coercion measure.
[c] Not a voter-verifiable e-voting scheme (pre-2015).

ignored, it is assumed (see Table 4.3) that the MNO is trusted to provide both the infrastructure and management procedures securely.

4.6.2 Formal Security Analysis

The Scyther protocol verification tool was used to formally analyse the generic SCWS e-voting protocol, and no attacks were found within bounds. The Scyther tool and its verification processes are fully described in Appendix B: the e-voting Scyther Script and the verification results obtained are shown in Appendix Section B.2.

4.7 Chapter Summary

This chapter set out the security requirements that remote e-voting systems must meet. A generic model for using a phone equipped with an SCWS/ SIM as a secure voter interface for e-voting was then presented. Two e-voting schemes, PAV and Estonian I-voting, were then used with this generic model, presented as separate use cases *EV-1* and *EV-2* respectively.

The strengths of the proposed design are that it uses standardised protocols, hardware and communications to simplify its design and operation. The security of the standardised elements of the design has been extensively investigated by the expert community. Using existing tamper-resistant hardware (the SIM) with standardised features (SCWS) along with the MNO's FAP (via HTTPs), means that sensitive information can be protected at all times. The voting application is only available to the voter via an HTTPs connection from the mobile phone handset, and only authorised

parties can access the SCWS voting application via the MNO network. Distributing web server functionality to voters' SIMs means that there is no central web server to target, and so an attacker must compromise many phones to successfully affect the election result. The use of the SIM's tamper-resistant environment for the storage and processing of sensitive voter credentials also addresses the secure platform problem. However, some of the specified security requirements were not met in use case *EV-2: SCWS I-voting* because the pre-2015 version of this voting system is not voter-verifiable: it is not possible to be sure that a vote will be counted correctly when received by the Voting Authority. The security of both use cases was assessed informally against the previously defined security requirements and the generic model was formally analysed using the protocol checking tool Scyther, and the results were promising. The principle of using a ubiquitous device (the phone) with SCWS to provide a secure distributed architecture for remote e-voting has been established. The voter will have a flexible and convenient "vote-anywhere" capability in their phone, whilst the e-voting system is protected by making the effort required to attack it prohibitively high.

A summary of how well the security requirements were met for each use case is shown in Table 4.7.

4.8 Related Publications

Two publications resulted from the work described in this chapter:

1. L. Kyrillidis, S. Cobourne, K. Mayes, S. Dong, and K. Markantonakis, "Distributed e-Voting using the Smart Card Web Server", in *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS2012)*, IEEE, 2012, pp. 1–8.[1]

2. S. Cobourne, L. Kyrillidis, K. Mayes, and K. Markantonakis, "Remote e-Voting Using the Smart Card Web Server", *International Journal of Secure Software Engineering (IJSSE)* vol. 5, no 1, pp.39–60, 2014.[2]

Chapter 5

Mobile Payments

Contents

In remote areas of developing countries, the mobile phone network may be the only connection with outside organisations such as banks, financial institutions or humanitarian aid providers. This chapter proposes two mobile payment applications: MP-1, a branchless banking system for withdrawal, deposit and transfer transactions, which uses an SCWS application on a SIM with public key cryptography capabilities; and MP-2, an m-payment system which uses basic feature phones to interface with online hosted Bitcoin Wallets maintained by a charity. Here, SMS messaging is used along with an OTP authentication token to enhance security.

5.1 Mobile Payment Use Cases

As previously discussed in Chapter 2, there are security concerns with many existing m-payment solutions. Two m-payment use cases are now described, *MP-1: SCWS Branchless Banking* and *MP-2: Bitcoin SMS m-Payment*, which propose security improvements on existing m-payment schemes.

Use case *MP-1: SCWS Branchless Banking* uses a PKI-capable SCWS/SIM in a branchless banking scheme, catering for both cash-based withdrawal/ deposit transactions via a network of authorised bank agents as intermediaries, similar to M-PESA. (An M-PESA agent is shown in Figure 5.1) . Transfers to third parties (non-agents) are also included in the scheme. The proposed solution provides security without requiring the customer to obtain expensive equipment or specialised software: all that is required is an existing phone handset (complete with a standard browser) which can have an advanced SIM installed.

Figure 5.1: M-PESA Agent: "M-PESA agent in Kibera, Nairobi, Kenya" by Fiona Graham / WorldRemit is licensed under CC BY-SA 2.0, accessed 23 May 2017. https://www.flickr.com/photos/worldremit/33322696760/

Use case *MP-2: Bitcoin SMS m-Payment* presents a new philanthropic model whereby charities can receive donations in Bitcoin: the security of the distributed ledger approach (blockchain) means that donations can be quickly, cheaply and transparently transferred to the charity's operations in the field. An SMS m-payment scheme is described which allows these Bitcoin donations to be used by beneficiaries in a humanitarian aid setting where there is limited internet availability. One Time Password (OTP) security tokens are employed to enhance the authentication of SMS transac-

tions, and the solution is a pragmatic balance between security and usability in an environment where access to Bitcoin is not normally possible.

The security requirements that need to be met by m-payment schemes are summarised in the next section.

5.2 M-Payment Security Requirements

An m-Payment system should meet the security requirements of confidentiality, integrity, authentication, availability and non-repudiation as shown in Table 5.1.

Table 5.1: Mobile Payments: Security Requirements

	Confidentiality
MP-SR1	Sensitive information should not be disclosed to unauthorised parties, whether during processing, in transit, or at rest.
	Integrity
MP-SR2	Information must not be tampered with by unauthorised parties when it is in transit or at rest
MP-SR3	The system must perform its tasks without unauthorised manipulation
	Authentication
MP-SR4	All participants in a transaction must be authorised
MP-SR5	All transaction data must be genuine
	Non-repudiation
MP-SR6	None of the participants in a transaction can subsequently deny taking part in it
	Availability
MP-SR7	A service is not denied to authorised entities: for example, through network connectivity problems, loss of equipment such as phones, or distributed denial of service (DDoS) attacks

The next section describes the first m-Payment use-case, *MP-1: SCWS Branchless Banking*, where a PKI-capable SCWS/SIM is used as the basis of a branchless banking scheme that enables cash-based transactions to be done with bank agents, with the option of transferring funds to third parties.

5.3 MP-1: SCWS Branchless Banking

5.3.1 Branchless Banking - Background Information

In any financial system, authentication of participating entities is vital for security. Existing branchless banking schemes use combinations of SMS, USSD and IVR mechanisms to communicate financial and authentication data between parties. For example,

Table 5.2: Customer Authentication Methods in Branchless Banking Schemes

Scheme	**Authentication Mechanism**
M-PESA	PIN sent via USSD with proprietary encryption
EKO	6-digit printed nonces combined with user's 4-digit PIN
ALW	Voice/Fingerprint Biometrics
FSB	Voice Biometrics plus scratch-card nonces
M-ATM	PIN/phone no. generate key for encrypted SMS, also SMS Nonces
mChek[a]	6-digit PIN, IVR and SMS One Time Password (OTP)

[a] Reports that mChek were in difficulties surfaced in 2012 [255], and the company is now closed.

M-PESA (Kenya) [48], uses two-factor authentication (i.e. possession of a phone and knowledge of a PIN) with USSD and proprietary security via a SIM Toolkit (STK) application [29], with SMS messages for transaction data: customers are issued with a SIM containing the M-PESA application. Other branchless banking schemes include: EKO Bank (India) [49], ALW/ZMF (India) [50], FSB (not yet deployed) [51], M-ATM (Sri Lanka) [52], and mChek (India) [53]. Their authentication methods are shown in Table 5.2.

SIM-based applications have many desirable security properties. Access to the SIM is tightly controlled, so it is difficult for malware on a phone to affect a SIM application. The SIM environment is tamper-resistant, which protects against physical attacks. If the SIM is able to perform public key cryptographic operations (PKI-capable) it can use standardised security algorithms [210, 211], which is a useful security feature for branchless banking.

5.3.2 MP-1: Security Requirements

In a branchless banking system the security requirements detailed in Section 5.2 apply to to all messages sent to/from the bank, and all information stored on agent and customer equipment. A customer needs assurance that the agent is genuine and authorised to deposit their money in the correct account, an agent must make sure that the individual withdrawing money from an account is not an impostor, and all bank-originating messages must be authenticated. Network connectivity problems, loss of equipment such as phones, or DDoS attacks should not affect the security of the system, and the agent, customer or bank should not be able to deny a transaction took place.

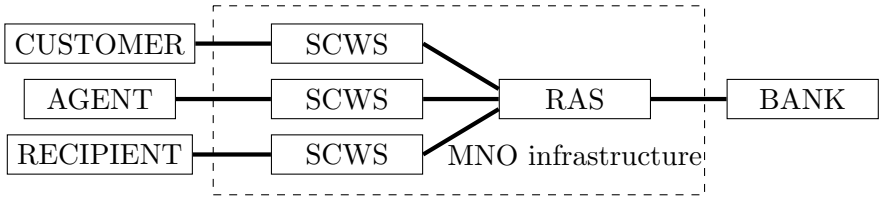

Figure 5.2: MP-1: SCWS-Banking - Entity Diagram

5.3.3 Using the SCWS for Branchless Banking

Modern SIMs can have advanced features in addition to the SCWS, such as the ability to perform public-key cryptographic processing (PKI-capable SIM), using standardised algorithms for encryption/ digital signatures [210, 211].

The use case presented now employs a PKI-capable SCWS/SIM to provide security in m-payment transactions without requiring the customer to obtain special equipment or software. The customer visits an authorised agent to withdraw/ deposit funds, and can also make transfers to third parties by harnessing the desirable security features of the SCWS and its tightly managed, standardised communications. Bringing web server functionality to the SIM environment means that users will be able to access web content offline using standard phone browsers. Users will also benefit from a richer interface which can incorporate files, images and multimedia as required [228]. This is particularly helpful for illiterate users, where graphical and voice based interactions are more effective than text-based menu-style SIM-toolkit applications [36], or for the visually challenged [256].

For brevity, this proposal will be referred to as SCWS-Banking throughout this chapter: the required entities and assumptions are shown in Table 5.3 and Table 5.4 respectively. The relationship between entities is illustrated in Figure 5.2[1].

The next sections will present SCWS-Banking transaction protocols: withdrawals, deposits and transfers. For simplicity, it is also assumed in the following descriptions that: if any of the protocol validation checks fail an error message is sent to all participants, the transaction is terminated and logged as unsuccessful; if the transaction cannot be completed for any reason e.g. due to lack of connectivity, a suitable rollback mechanism is used to reverse any partially completed processing; all cryptographic keys are checked for validity before use; data is padded according to best practice recommendations before being encrypted using a standardised public key algorithm e.g.

[1]Referring to the business models described in Section 3.2.5, a TSM could control updates to the SCWS on behalf of the bank, using delegated or authorised mode. In certain regulatory environments, the MNO can act as the bank by storing value on behalf of the customer: the mobile phone number is the account number, as in M-PESA [48]. Here, the simple mode business model would be appropriate. In this use case *MP-1*, the term "bank" will be used for both MNO-centric and bank-centric scenarios.

Table 5.3: MP-1: SCWS-Banking - Entities

Entity	**Description**
Bank (B)	The Bank processes all financial transactions, and maintains central databases of customer/agent accounts. It uses the procedures outlined in Chapter 3 to install its banking application and relevant security credentials on customers' and agents' SCWS/SIMs
Agent (A)	An agent is authorised to process transactions on the bank's behalf.
Customer (C)	The customer is an individual who performs financial transactions.
Recipient (R)	The recipient is an individual who receives transferred value from a customer
MNO	The MNO provides the technical mobile infrastructure and standardised SCWS administration protocols (see Chapter 3). The MNO provides a managed space on the SCWS/SIM for the bank's exclusive use, as described in [257].
Phone	The mobile phone handset and browser application are generally untrusted, as jail-broken operating systems or malware can compromise the correct operation of the phone's applications and operating system.
RAS	The RAS is a mere conduit between the bank and participants' SCWS/SIMs: it may be part of an MNO, TSM or bank, and passes messages unaltered to/from the bank's transaction processing system, with additional phone/SIM routing information as required. The RAS communicates with each SCWS using FAP/HTTPs sessions: however as HTTPs is not running end-to-end throughout a whole transaction, application level security mechanisms are included to prevent confidential information being visible at the RAS.
SCWS	A Java applet running on the SCWS will use relevant credentials and keys present on the SIM, and create dynamic content whenever requested: this applet will be referred to as the SCWS-Banking application throughout this chapter.

Table 5.4: MP-1: SCWS-Banking - Assumptions

	Description
MP-1-A1	**Registration:** agents and customers register with the bank, when appropriate identity documents are checked to satisfy banking regulations(e.g. Know Your Customer (KYC), Anti-Money-Laundering (AML) and Countering the Financing of Terrorism (CFT)). In some regulatory environments agents can check and register customers [258]: in others, customers/agents must go to the bank to register. Customers' identity details are stored by the bank for later use. Agents are allocated an Agent ID to display publicly. Customers/agents are issued with SCWS/SIMs, containing the SCWS-Banking application and their account credentials
MP-1-A2	**Banking Credentials on SCWS:** these are: an SCWS PIN (passwords may not be suitable for illiterate customers [36, 35]); two customer public/private key pairs (for key separation purposes, one pair for encryption/decryption and one pair for signing/verifying), with key sizes following recommended guidelines e.g. [241]; and two bank public keys, for encrypting/verifying messages to/from the bank.
MP-1-A3	**Availability of Equipment and Services:** it is envisaged that the customer will possess a mobile handset with a browser, but if necessary their SCWS/SIM could be inserted in a shared phone to access SCWS-Banking. An agent must have a phone with SCWS/SIM. It is assumed a mobile phone network is available, although connectivity could be intermittent.
MP-1-A4	**Access to SCWS-Banking System:** customers and agents participating in an SCWS-Banking transaction must first authenticate themselves to the SCWS environment by inputting a PIN to the phone browser.
MP-1-A5	**Account Structure:** there is a one-to-one correspondence between a SCWS and a bank account number: this means that an SCWS mobile phone number can be used to uniquely identify a customer or agent.
MP-1-A6	**Trust:** the customer does not trust the agent, and vice-versa. The bank and RAS are fully trusted.
MP-1-A7	**Bank/MNO relationship:** there is a trusted one-to-one relationship between the bank and MNO: i.e. a specific bank will partner with one MNO only.

Table 5.5: MP-1: SCWS-Banking - Protocol Notation

Notation	Description
A	Agent(entity)
AC_X	Account Number for entity X
B	Bank(entity)
BAL_X	Balance in Account AC_X for entity X
BAL'_X	Updated Balance in Account AC_X for entity X
C	Customer(entity)
CH_X	Result of identity check for entity X, value = *true/false*
E_K(Z)	Encryption of data Z with key K
ID_X	Identity of entity X
N_X	Random Nonce generated by entity X
$NAME_X$	Name of entity X, (i.e. a short identifying text)
Ph_X	Phone Number of entity X
PK_X/ SK_X	Public/ Secret Key pair of entity X
R	Recipient(entity)
S_X/ V_X	Signing/ Verification key pair of entity X
Tr	Transaction Type:'W'=Withdrawal, 'D'=Deposit, 'T'=Transfer
TrAmt	Transaction Amount
$TrCount_X$	Transaction Counter for entity X
TrNo	Transaction Number
X→Y:	Message sent from entity X to entity Y
(Z)$Sign_K$	Signature on data Z with signature key K

RSA [210]; and a standardised digital signature algorithm is used e.g. DSA [211]. The notation used is shown in Table 5.5.

5.3.4 SCWS-Banking Withdrawal Protocol

In a withdrawal, the customer enters transaction details, the bank authorises them and forwards them on to the agent to authorise in the presence of the customer. Figure 5.3 shows the messages in a withdrawal transaction.

Step 1: Customer enters *TrAmt*, ID_A: The SCWS-Banking application generates N_C, increments $TrCount_C$ and creates message W1 using PK_B, S_C to encrypt and sign. The SCWS triggers a FAP session, the RAS retrieves Message W1 (over HTTPs), adds Ph_C and passes it on to the bank.

W1 C→B: $(E_{PK_B}(Tr, TrAmt, ID_A, N_C, TrCount_C))Sign_{S_C}$

Step 2: Bank authorises transaction: the bank uses Ph_C to find AC_C, BAL_C, V_C, PK_C, $NAME_C$ and ID_C, and verifies/ decrypts Message W1 using relevant keys. The bank checks $TrCount_C$, and uses ID_A to obtain Ph_A, BAL_A, V_A and PK_A. If

$TrAmt \leq BAL_C$, the bank generates N_B and $TrNo$, creates Message W2 (encrypted/ signed with PK_A/ S_B) and sends it to the RAS with Ph_A. The RAS uses Ph_A to forward message W2 to the agent via FAP/HTTPs.
W2 B→A: $(E_{PK_A}(Tr, TrNo, TrAmt, NAME_C, ID_C, N_B, BAL_A))Sign_{S_B}$

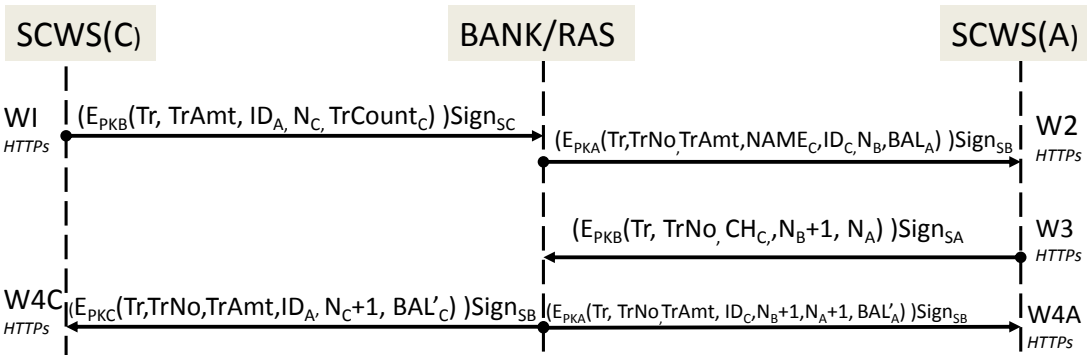

Figure 5.3: MP-1: SCWS-Banking - Withdrawal Protocol

Step 3: Agent authorises transaction: the agent SCWS-Banking application verifies/decrypts message W2 using V_B/ SK_A, then checks that $TrNo$ has not been received before. The agent inputs ID_C, and the SCWS-Banking application checks if ID_C(input) = ID_C(from W2) and sets CH_C: N_B is incremented, N_A is generated, and a transaction log is updated. Message W3 is created, retrieved via RAS/FAP/HTTPs, the RAS adds Ph_A and passes it to the bank.
W3 A→B: $(E_{PK_B}(Tr, TrNo, CH_C, N_B + 1, N_A))Sign_{S_A}$

Step 4: Bank finalises and confirms transaction: the bank uses Ph_A to obtain agent keys to verify/decrypt message W3. The bank inspects $N_B + 1$ and CH_C: if the $CH_C = true$, $TrAmt$ is used to create BAL'_A and BAL'_C. The transaction is logged, then time-stamped confirmation messages are sent to the agent/customer via SMS, and (encrypted and signed) to their SCWS-Banking applications, via RAS/FAP/HTTPs (messages W4A and W4C).
W4A B→A:$(E_{PK_A}(Tr, TrNo, TrAmt, ID_C, N_B + 1, N_A + 1, BAL'_A))Sign_{S_B}$
W4C B→C: $(E_{PK_C}(Tr, TrNo, TrAmt, ID_A, N_C + 1, BAL'_C))Sign_{S_B}$

Step 5: Agent and customer finalise transaction: the SCWS-Banking applications verify/ decrypt message W4A or W4C (as appropriate) from the bank, update the SCWS-Banking files with transaction data, and the transaction is logged. The agent should only give the customer cash once the confirmation message has arrived

from the bank. A paper transaction log is also maintained by the agent, which the customer must sign to acknowledge receipt of the cash.

5.3.5 SCWS-Banking Deposit Protocol

A deposit is an agent-initiated transaction similar to a withdrawal, but with the message flow reversed. The deposit protocol messages are shown in Figure 5.4, and summarised below: again, phone numbers are added to messages between the bank and the RAS for routing purposes.

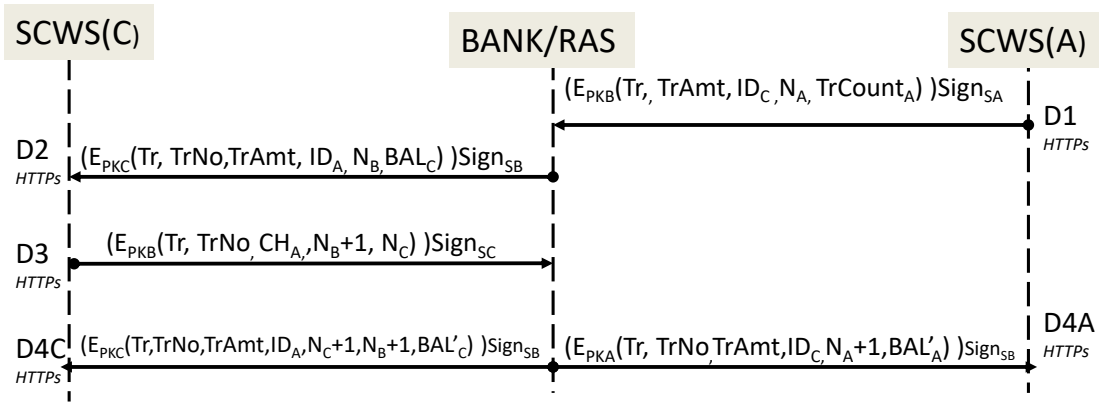

Figure 5.4: MP-1: SCWS-Banking - Deposit Protocol

Step 1: Agent inputs *TrAmt*, ID_C (Agent to Bank)
D1 $A \rightarrow B: (E_{PK_B}(Tr, TrAmt, ID_C, N_A, TrCount_A))Sign_{S_A}$
Step 2: Bank authorises transaction (Bank to Customer)
D2 $B \rightarrow C: E_{PK_C}(Tr, TrNo, TrAmt, ID_A, N_B, BAL_C))Sign_{S_B}$
Step 3: Customer authorises transaction (Customer to Bank)
D3 $C \rightarrow B$: $E_{PK_B}(Tr, TrNo, CH_A, N_B + 1, N_C))Sign_{S_C}$
Step 4: Bank finalises and confirms transaction
D4A $B \rightarrow A: E_{PK_A}(Tr, TrNo, TrAmt, ID_C, N_A + 1, BAL'_A))Sign_{S_B})$
D4C $B \rightarrow C: E_{PK_C}(Tr, TrNo, TrAmt, ID_A, N_B + 1, N_C + 1, BAL'_C))Sign_{S_B})$

5.3.6 SCWS-Banking Transfer Protocol

The bank transfers value *TrAmt* from a customer to a recipient (R), directly if the recipient's account is known, otherwise via an SMS to the recipient for redeeming *TrAmt* from an agent later. Transfer messages are shown in Figure 5.5.
Step 1: Customer inputs *TrAmt*, Ph_R (Customer to Bank)
T1 $C \rightarrow B$: $(E_{PK_B}(Tr, TrAmt, Ph_R, N_C, TrCount_C))Sign_{S_C}$

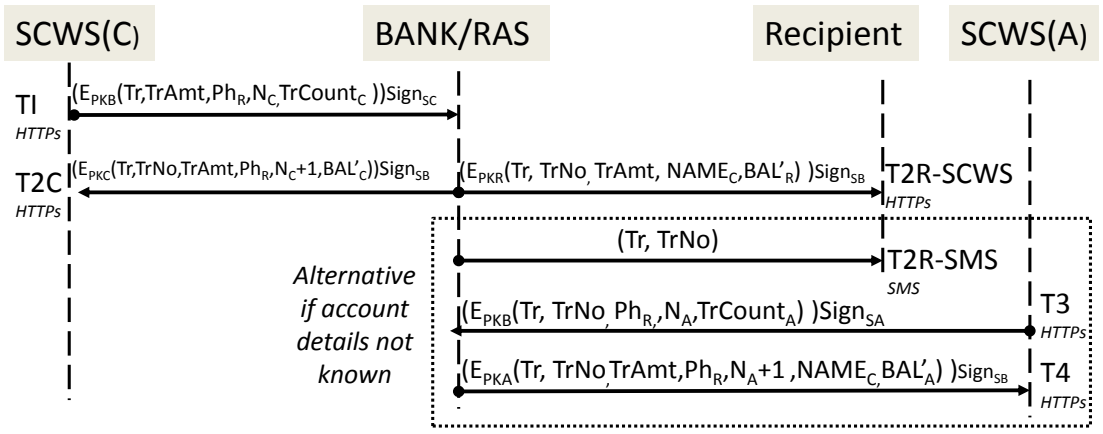

Figure 5.5: MP-1: SCWS-Banking - Transfer Protocol

Step 2: Bank processes transaction (Bank to Customer/ Recipient)
T2R-SMS is sent via SMS to Ph_R if the recipient does not have a SCWS-Banking account: Steps 3 and 4 are also needed in this case.
T2C B→C: $(E_{PK_C}(Tr, TrNo, TrAmt, Ph_R, N_C + 1, BAL'_C))Sign_{S_B}$
T2R-SCWS B→R: $(E_{PK_R}(Tr, TrNo, TrAmt, NAME_C, BAL'_R))Sign_{S_B}$
T2R-SMS B→R: (Tr,TrNo)
Step 3: Recipient redeems SMS transfer (Agent to Bank)
The recipient gives the agent Ph_R and *TrNo*.
T3 A→B:$(E_{PK_B}(Tr, TrNo, Ph_R, N_A, TrCount_A))Sign_{S_A}$
Step 4: Bank confirms transaction to agent (Bank to Agent)
The bank obtains full transaction details from its records and instructs the agent to pay the recipient *TrAmt*. $NAME_C$ can be given to the recipient for their records. The agent should also manually record the recipients ID credentials, and maintain a paper transaction log for non-repudiation purposes, which the customer must sign as acknowledgment.
T4 B→A:$(E_{PK_A}(Tr, TrNo, TrAmt, Ph_R, N_A + 1, NAME_C, BAL'_A))Sign_{S_B}$

A preliminary security analysis of the SCWS-Banking scheme will now be done.

5.3.7 Security Analysis

The security of the SCWS installed on the tamper-resistant SIM is described in more detail in Chapter 8. The SCWS-Banking proposal is now discussed with respect to the security requirements set out in Section 5.2. Potential attacks and mitigations are identified and a comparison with the SMS-banking scheme M-PESA is also shown.

- **Confidentiality (MP-1-SR1):** All information sent between the SCWS, RAS, and the phone browser are protected by HTTPs against eavesdropping and man-in-the-middle attacks whilst in transit. As HTTPs is not running end-to-end - there are separate RAS/FAP/HTTPs sessions for each message - application level security is also used to meet security requirements: public key encryption and the PIN-protected tamper-resistant SCWS environment ensures that sensitive information is kept confidential at all times.

- **Integrity (MP-1-SR2/MP-1-SR3):** Using HTTPs between the RAS/ SCWS/ phone browser gives reasonable assurance that information sent is not tampered with. Messages are digitally signed to allow detection of unauthorised modifications, and challenge-responses prevent replay attacks. Replay attacks where messages W1/D1/T1 are recorded and subsequently resent to the bank to generate multiple transaction authorisation numbers (*TrNo*) are prevented by the use of transaction counters held on the SCWS and checked by the bank. The tamper-resistant SIM makes attacks on data integrity extremely difficult.

- **Authentication (MP-1-SR4/ MP-1-SR5):** ID credentials are input by the customer/agent and verified by the bank, so imposters will be identified. Digital signatures are used in all messages for assurance that the sender is genuine, and the RAS and SCWS authenticate each other using HTTPs. The bank is authenticated by the use of digital signatures, and it sends confirmations via two separate channels (SMS/ FAP to SCWS).

- **Availability (MP-1-SR6):** Network connection problems encountered during a FAP session are automatically handled by the SCWS on-card Administration Agent (see [15]). Data held on the SCWS is available offline, and is only accessible by using a PIN. Back-office procedures are needed for remote locking/ reissue of the SIM/SCWS application and credentials when phones/SIMs are lost/damaged. The SCWS-Banking application and credentials are installed on many SIMs, so DDoS attacks are hard to mount because each phone has to be targeted individually and the attacker has to be in possession of the phone. A DoS attack against a single SCWS is feasible but difficult and not scalable. The RAS is potentially a single point of failure in terms of availability, as it controls the entire message flow in the SCWS-Banking protocol. However, as it is a trusted entity, owned and operated by the MNO, it is subject to tightly controlled management procedures which should make unauthorised usage and attacks difficult.

- **Non-Repudiation (MP-1-SR7):** Digital signatures provide non-repudiation;

Table 5.6: M-PESA /SCWS-Banking Security Comparison

SR	M-PESA	SCWS-Banking
Confidentiality (MP-1-SR1)	Unencrypted SMS messages, readable from phone's SMS inbox	Messages are encrypted, data PIN protected on SCWS
Integrity(MP-1-SR2/3)	Malware could intercept and tamper with SMS messages on phone	SMS used for confirmation only
Authentication (MP-1-SR4/5)	Bank-originating messages not authenticated	Digital Signatures are used for authentication
Availability (MP-1-SR6)	Large number of SMS messages could flood mobile network: needs lost/stolen phone/SIM procedures	DDoS resistant: needs lost/stolen phone/SIM procedures
Non-repudiation (MP-1-SR7)	Message from bank not authenticated, so no non-repudiation	Digital signatures used for non-repudiation

additionally, transaction logs are securely held on both the agent/customer SCWS and centrally by the bank. Paper-based transaction logs are maintained by the agent and signed by the customer to acknowledge each transaction. Additionally, the bank sends confirmation messages via two channels, to minimise the likelihood of losing a message in transit.

Comparison with M-PESA SMS Scheme: Table 5.6 compares the security of SCWS-Banking with M-PESA. It can be seen that M-PESA only partially meets all the identified security requirements, whereas the proposed SCWS-Banking solution satisfies them all.

5.3.8 Formal Security Analysis

The Scyther protocol verification tool was used to formally analyse the withdrawal, deposit and transfer protocols proposed for use case *MP1: SCWS Branchless Banking*, and no attacks were found within bounds. The Scyther tool and its verification processes are fully described in Appendix B, and the Scyther Scripts and verification results obtained are shown in Appendix B.3.

5.3.9 Use Case MP-1: Summary

An SCWS-Banking scheme has been presented that uses PKI-capable SIMs equipped with a SCWS to process branchless banking withdrawals, deposits and transfers in

a secure and user-friendly manner. The main strength of the proposal is that it uses standardised hardware, protocols and communications to protect sensitive information, without the need for specialised equipment and phone applications: all communications to/from the SCWS are done via HTTPs. By storing security information on the tamper-resistant SIM, local authentication of PINs can be done by the SCWS without communicating credentials across a network. All transactions pass through the trusted RAS, owned and operated by the MNO or a TTP. It is difficult to mount large scale attacks against the system, as credentials and applications stored on each SIM must be targeted individually. PKI-capable SIMs enable application level public key encryption/ digital signatures to provide authentication and non-repudiation, using keys stored on the tamper-resistant SCWS/SIM. Agents and customers need new advanced SIMs containing the SCWS-Banking application and their account credentials: even though these are more expensive than conventional SIMs, this could be a cheaper overall solution than setting up physical bank branches. A preliminary security analysis indicates that the security of SCWS-Banking is higher than that offered by M-PESA. The initial findings are promising, and the SCWS-Banking proposal meets branchless banking security challenges very well.

We now move on to an m-Payment system proposal *MP-2: Bitcoin SMS m-Payment* designed for use in humanitarian aid scenarios, that uses SMS messaging to interface with Bitcoin wallets hosted by a charity.

5.4 MP-2: Bitcoin SMS m-Payments

5.4.1 Bitcoin for Charity - Background Information

This use case explores how blockchain technology can be used to provide financial services in humanitarian aid scenarios, where there may not be Internet connectivity. Full details of the proposed solution can be seen in the paper published as a result of this work [4]: the Bitcoin processing in the protocol is described in Appendix A.3. The following account is a summary that focuses more on the "last-mile" aspect of the proposal i.e. using SMS messaging for financial transactions.

Bitcoin is a decentralised cryptocurrency system which works on a peer-to-peer network. Payments are made to Bitcoin addresses (Bitcoin public keys) which can be generated using an Elliptic Curve Digital Signature Algorithm (ECDSA) [259, 260]. A Bitcoin transaction transfers monetary value attached to a particular Bitcoin address by digitally signing a hash of the previous transaction together with the next owner's Bitcoin address, thereby creating a chain of signatures that links past and present transactions. Transactions are permanently recorded in files called blocks, that are

timestamped and chained together in the order they appear (the "blockchain", also referred to as a globally distributed cryptographic ledger), and shared/ synchronised with all the nodes connected to the peer-to-peer network. The correctness of the blockchain underpins the security of Bitcoin.

There has been much interest in alternative ways blockchain technology could be used in the philanthropic sector, for example to increase transparency, openness and trust whilst reducing transaction costs and providing new opportunities for fundraising [261, 262, 263, 264]. For example, the Royal National Lifeboat Institution (RNLI) has accepted Bitcoin since August 2015, hoping to attract new donors from a different demographic in addition to its typical supporters [265]. Also, the BitGive Foundation's donation tracking service (GiveTrack) allows donors to trace Bitcoin transactions in real time, showing how donations are spent [266].

Blockchain based schemes have fundamental technical requirements such as Internet connectivity and compatible devices that can perform the required cryptographic processes needed for a Bitcoin transaction (e.g. secure hash generation, digital signatures and secure storage of cryptographic keys). There are organisations that keep the most up-to-date blockchain to verify and forward transactions on behalf of registered users, via online wallets (also known as hosted wallets). However, to make a payment, the user must be online to access their wallet via a web browser or smart phone application. In the proposal in this use case the charity can set up hosted wallets, allocate them to beneficiaries then transfer financial aid directly.

5.4.2 SMS m-Payment Schemes and Bitcoin

If there is no reliable Internet facility available, the online hosted wallet approach for blockchain transactions is not a practical option. However, an SMS-based solution could interface with a hosted wallet back-end system.

As described in Chapter 2, SMS m-payment systems have been extremely successful in the developing world, most notably M-PESA in Kenya [212]. There are SMS-based m-payment schemes available that facilitate Bitcoin transactions, but they have met with varying success. None of the existing Bitcoin SMS solutions described in Section 2.5 is suitable for the offline environment under discussion, as they all require the user to have some degree of online access to their Bitcoin Wallet. This use case presents an m-payment system that uses SMS messages to transact with Bitcoin wallets hosted on beneficiaries' behalf by a charity. Offline beneficiaries can then make and receive Bitcoin payments using SMS messaging on basic feature phones. A Hash-based OTP (HOTP) security token is included to provide some assurance that only a genuine user

can send an SMS to make a transaction[2]. More information about OTPs can be seen in Appendix A.1.1. All SMS messages used in the proposal are within the standard 160 character length.

There are two Bitcoin payment processing methods that are of interest for this use case: Multi-Signature Addresses and Smart Contracts. *Multi-Signature Addresses* are derived using a multi-signature process, where more than one private key is needed to authorise a Bitcoin transaction. For example, a 2-of-3 multi-signature is when a Bitcoin address is associated with three private keys and at least two out of the three private keys are needed to authorise a Bitcoin transaction. In our proposal, "Pay To Script Hash" (P2SH) transactions are used to process multi-signatures. To generate a multi-signature, a Full Redeem Script which includes details of the three public keys is hashed to generate a hashed Redeem Script which becomes the P2SH multi-signature. The Full Redeem Script is shared between all key-holding entities. The Redeem Script can be used to verify the transferred amount and whether its being sent to the correct multi-signature address. It also gives details about how many signatures are needed to make a payment. The recipient needs to provide the full Redeem Script to spend the received Bitcoins. Some services use this technique to enhance security of hosted Bitcoin wallets e.g. Bitgo [267]. *Smart Contracts* are coded instructions published on a distributed network, that can receive inputs, execute instructions and provide outputs. The multi-signature scheme (Option 1) can be classed as a very low level smart contract that can only process simple Bitcoin payment transactions. A smart contract could enable a charity to extend the services it offers: e.g. issuing aid to beneficiaries on a regular or ad-hoc basis, small micro-finance loans, keeping of repayment records, automatic communications with donors (such as donation requests/audit reports). There have also been proposals to use blockchain smart contacts for identification of refugees [268]. However, as the Bitcoin blockchain was initially designed as a distributed payment platform, it is not possible to run advanced smart contracts on it, so alternative platforms such as Rootstock (RSK)[269] are more suitable.

5.4.3 MP-2: Security Requirements

The security requirements detailed in Section 5.2 apply to to all messages sent between participants in a transaction, and all information stored on mobile devices. All parties need assurance that the counter-party in a transaction is not an impostor, and all messages that originate from the charity/aid organisation are genuine. Network con-

[2]Time-Based (TOTP) tokens generate new codes automatically after a set period of time: this approach is not suitable for use with SMS messages that may be subject to potential delays in the messaging system.

nectivity problems, and DDoS attacks should not affect the security of the system, and no participant in a transaction can subsequently deny it took place. An adversarial model relating to a humanitarian aid setting has been described in [270].

The use of SMS messaging to transport financial transaction data is not ideal: the SMS system was not designed with security in mind, and well known attacks such as spoofing and Man-in-the-Middle may compromise the stated security requirements. However, in an environment where this is the only communication option, the proposed solution will need to strike a pragmatic balance between security, usability and practicality.

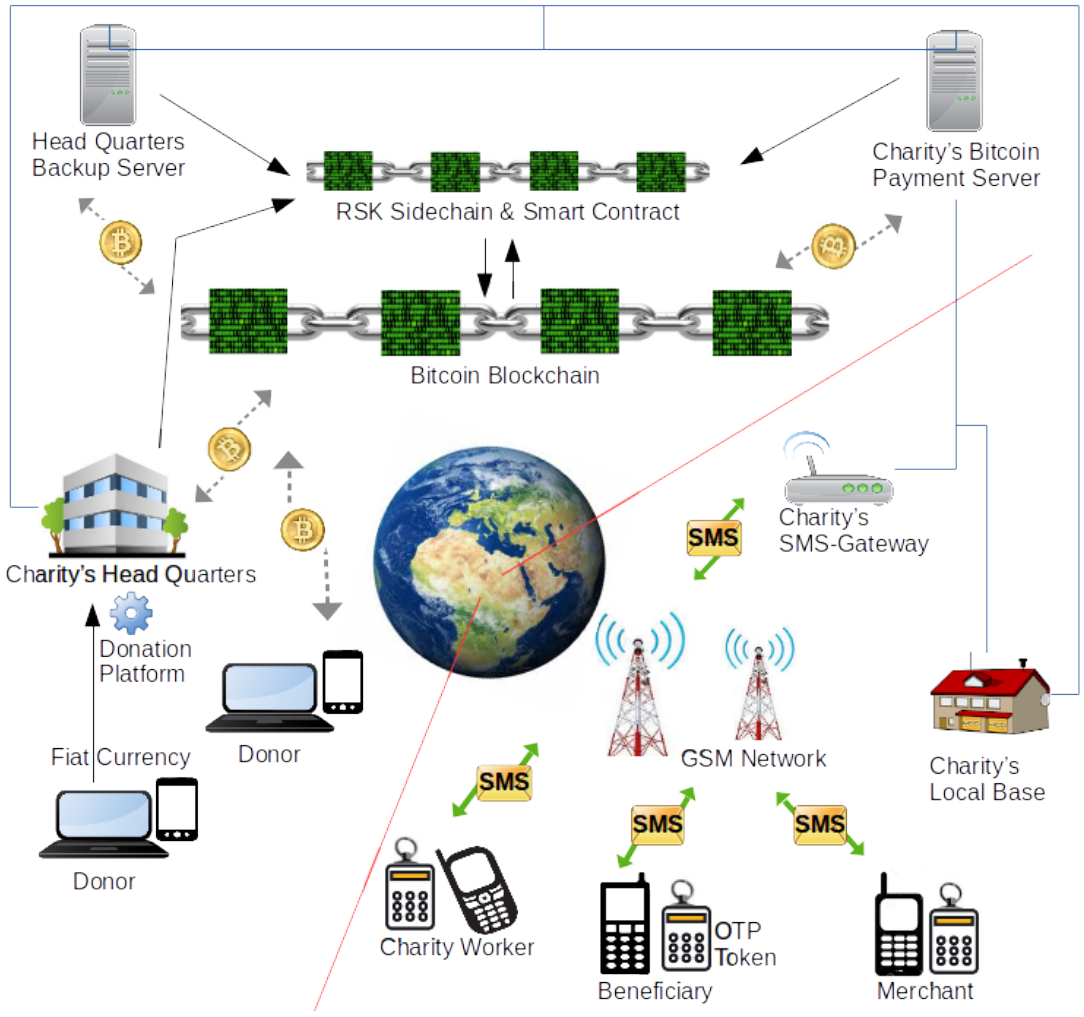

Figure 5.6: MP-2: Bitcoin SMS Payments - System Architecture

The next section describes the proposed method to use SMS messaging in a Bitcoin m-payment scheme.

Table 5.7: MP-2: Bitcoin SMS Payments - Entities

Entity	**Description**
Bitcoin Payment Server (BPS)	This is part of the charity's technical infrastructure, and manages hosted Bitcoin Wallets on behalf of beneficiaries. It securely holds Bitcoin keys for each account holder, and is connected to the Bitcoin/RSK peer-to-peer network. It also checks and signs payment requests received from the SMS-Gateway, and once these have been authorised by one of the other key holders, the BPS broadcasts them to the Bitcoin peer-to-peer network.
Blockchain	The distributed ledger shared between the nodes connected to the Bitcoin peer-to-peer network.
Charity Local Office (LO)	The Charity has a local office in the disconnected environment: the LO registers phone numbers of users, and manages distribution of OTP tokens.
Charity Head Quarters (HQ)	The Charity HQ may be geographically distant from the aid environment, and has online access/ secure servers: the HQ holds relevant Bitcoin private keys for all payers.
OTP Token	This is a cheap Hash-based One Time Password (HOTP) security token used with every SMS transaction. The algorithm that is used to generate the OTP is synchronised between the BPS and each individual security token. Sample OTP generation algorithms can be found in [271, 272]. Note: this token could be replaced by a SIM Toolkit application if the charity has a business relationship with the MNO
SMS-Gateway	This is a server that sends and receives SMS transmissions to and from the telecommunication network, and is connected to the BPS.
Donor Platform	Donors select Bitcoin addresses from a web based donor platform, and can use a Bitcoin wallet/client or fiat currency to donate.
Payer/ Recipient	Users of the system can make payments (Payer) to any other registered user (Recipient).

5.4.4 MP-2: Using SMS with Bitcoin Hosted Wallets

The charity creates hosted wallets for beneficiaries, and during a secure registration process at the local office, issues OTP tokens that will be used with each transaction. The proposal involves interactions between a number of entities, described in Table 5.7: the relationship between entities is illustrated in Figure 5.6. Necessary assumptions are shown in Table 5.8.

Table 5.8: MP-2: Bitcoin SMS Payments - Assumptions

	Description
MP-2-A1	**Charity Head Quarters (HQ):** The charity operates on an international level while providing humanitarian aid for offline beneficiaries. Revenue can come from donations made via a web-based donor platform. The charity is a reputable and trusted entity, with secure premises and online access/ backup servers which may be geographically distant from the aid environment.
MP-2-A2	**Donors:** Potential donors must have online access to use the donor platform.
MP-2-A3	**Bitcoin Payment Server (BPS):** The BPS is a secure server managed under industrial standard security controls and best practices to prevent attacks. All security keys are kept encrypted and stored securely to minimise the risk associated with data breaches.
MP-2-A4	**Phones:** All users of the system possess simple mobile phones (feature phones) that are protected by security code/access PINs, and the local existing GSM network can be used for SMS messages.
MP-2-A5	**Secure Registration:** During a secure registration process at the LO, the following procedures take place: 1) All users of the system register their mobile numbers and be issued with security tokens. 2) the mobile numbers and OTP security token IDs of users are sent to the BPS (encrypted using the LO's private key), in batches if the LO's internet connection is intermittent 3) mobile numbers are assigned an OTP identifier and Bitcoin wallet (stored online on the BPS).
MP-2-A6	**OTP Security Token:** This is a cheap hardware security token, used every time an SMS transaction is made, that generates HMAC-Based (HOTP) passcodes when the user requests ("event-driven"). These codes remain valid until used by the authenticating application. More information about OTP processes can be seen in Appendix A.1.1.
MP-2-A7	**Trust:** The SMS-Gateway and BPS are assumed to be trusted and secure. Mobile phones are not.

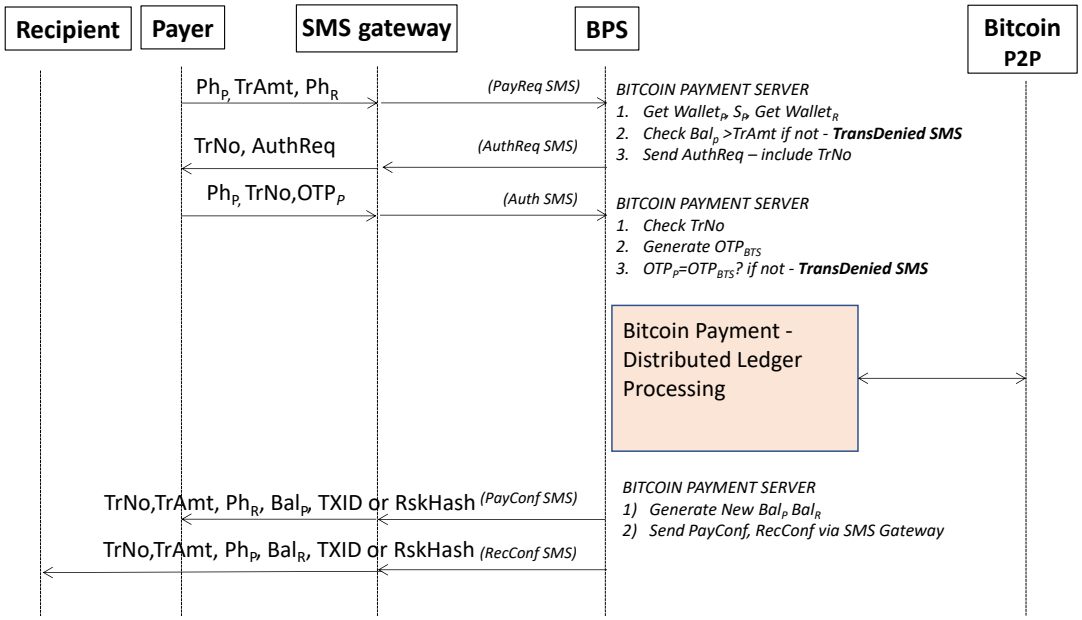

Figure 5.7: MP-2: Bitcoin SMS Payments - Protocol

5.4.5 MP-2: Bitcoin SMS Transactions

Payments can be made from charity worker to beneficiary, beneficiary to merchant, or merchant to merchant[3], and a summary of the message flow is shown in Figure 5.7[4]. The notation used is shown in Table 5.9, security credentials for each entity are shown in Table 5.10 and the content of each SMS messages used is shown in Table 5.11.

Stage 1: Payment Request To make a payment, the Payer (P) types an SMS message with payment instructions (*PayReq SMS*), and sends it to a local phone number provided by the charity, to be forwarded to the charity's BPS via the SMS-Gateway. The BPS retrieves Bitcoin wallets for both Payer and Recipient, checks *TrAmt* is not greater than BAL_P, pseudo-randomly generates a three-digit number, unique per transaction *TrNo*, and then sends *AuthReq SMS* asking for Payer's OTP. The Payer presses a button on the OTP token, then sends *Auth SMS* containing the resulting OTP to authorise the transaction. The BPS checks the TrNo, generates OTP_{BPS} and compares to the received OTP_P. If any checks fail, *TransDenied SMS* is sent to the Payer. If all checks are passed then the BPS proceeds to making a Bitcoin payment, using one of the two options described in Section 5.4.2.

[3]Merchants could use an existing Bitcoin address, registered and associated with a short Merchant ID by the BPS, used instead of Ph_P/ Ph_R in transactions.

[4]The two options for Bitcoin payment processing are described in full in Appendix A.3.1

Table 5.9: MP-2: Bitcoin SMS Payments - Protocol Notation

Notation	Description
$Addr_X$	Bitcoin Multi-signature Address for entity X
BPS	Bitcoin Payment Server(entity)
BAL_X	Bitcoin balance in Account AC_X for entity X
LO	Local Office (entity)
OTP_X	One Time Password generated by entity X
P	Payer(entity)
Ph_X	Phone Number of entity X
PK_X/ SK_X	Public/ Secret Key pair of entity X
R	Recipient(entity)
S_X/ V_X	Signing/ Verification key pair of entity X (Bitcoin keys)
TrAmt	Transaction Amount
TrNo	Transaction Number
TXID	Unique Transaction ID of a transaction recorded in the blockchain. Also referred to as the Transaction Hash (TrHash)
TrHash	Transaction Hash
X→Y:	Message sent from entity X to entity Y
RSKHash	Rootstock Transaction Hash

Table 5.10: MP-2: Bitcoin SMS Payments - Credentials

Entity	Keys and Other Assets
Payer/ Recipient	No keys, PIN for phone, HOTP token (no PIN) for making payments
BPS	S_{P-BPS},$Addr_{P-BPS}$, $Addr_{R-BPS}$, PK_{LO}, Ph_X, OTP_X
LO	S_{LO}, Physical OTP tokens, phone numbers (payers/recipients), plus registration details/ OTP allocation details
Donor	S_{Donor}/ V_{Donor}
Donor Platform	$Addr_{Project}$

Stage 2: Bitcoin Transaction Processing

The two options for Bitcoin transaction processing (i.e. Multi-Signature Addresses and Smart Contracts) are described in detail in Appendix A.3.1.

Stage 3: Payment Finalisation

Once the Bitcoin/RSK transaction has completed, the BPS updates the payer/recipient balances and sends confirmation messages via the SMS-Gateway: *PayConf SMS* or *PayConfRSK SMS* to the Payer and *RecConf SMS* or *RecConfRSK SMS* to the Re-

Table 5.11: SMS Payment Messages

Message	**Content**
PayReq SMS	Ph_P, TrAmt, Ph_R
AuthReq SMS	TrNo, AuthReq
Auth SMS	Ph_P, TrNo,OTP_P
TransDenied SMS	Ph_P, TrNo, Ph_R, Denied
PayConf SMS	TrNo,TrAmt, Ph_R, BAL_P, TXID
RecConf SMS	TrNo,TrAmt, Ph_P, BAL_R, TXID
PayConfRSK SMS	TrNo,TrAmt, Ph_R, BAL_P, RSKHash
RecConfRSK SMS	TrNo,TrAmt, Ph_P, BAL_R, RSKHash

cipient. TXID/RSKHash are included as unique IDs that can be used to trace the particular transaction on the Bitcoin/RSK blockchain if required.

5.4.6 Security Analysis

Security aspects of the proposed SMS payment scheme now analysed with respect to the m-Payment Security Requirements set out in Section 5.2. Potential attacks will also be identified.

- **Confidentiality (MP-2-SR1):** Bitcoin donors can remain anonymous if they choose, but this may introduce management issues for the charity. Some anonymous donations may need special reporting and investigation due to possible money laundering/ suspicious financial activity regulations. For example, in the UK, anonymous donations over £25,000 have to be reported as a "serious incident" [273]. To comply with these, a charity policy may be needed requiring identification for donations over a certain amount. Server attacks affecting HQ/HQB/BPS are possible; Table 5.12 shows recommended countermeasures. SMS messages are not encrypted by default, so there may be attacks on confidentiality.

- **Integrity (MP-2-SR2/MP-2-SR3):** Again, there may be server attacks directed at the HQ/HQB/BPS/ Donor Platform: Table 5.12 shows recommended countermeasures to ensure the integrity of information. SMS replay/spoofing attacks may occur, but the OTP in the *SMSAuth* message is designed to counter these attacks. If the charity has a business relationship with an MNO, the OTP token could be replaced with a SIM Toolkit application that generates OTPs and encrypts messages. However this would require beneficiaries to have a SIM with the application installed, which would reduce the ease of implementation of the scheme: a classic security versus usability dilemma.

Table 5.12: Attack Targets and Countermeasures

Attack Target	**Countermeasure**
Donor Platform	Platform is hosted on a secure web server adhering to industrial standard security controls to defend against external attacks such as DDoS, website defacing, content manipulation
HQ/HQB/BPS (DDoS)	HQ/HQB has secure premises and backup servers: BPS managed under industrial standard security controls and best practices to prevent attacks.
HQ/HQB/BPS (privilege escalation)	Server related attacks can be prevented by using security controls such as: access control, routine web-application vulnerability assessment/patching. Data breaches mitigated by securely storing security keys encrypted
SMS (MNO/GSM)	GSM/SMS security issues partially mitigated by the use of the OTP security token and TXID/RSKHash on confirmations
SMS spoof	OTP/TXID/RSKHash gives some assurance that payment is genuine
SMS replay	OTP prevents replay attacks
Blockchain/RSK (DDoS)	DDoS attacks not viable in distributed ledger, and integrity is innate in blockchain solutions

- **Authentication (MP-2-SR4/ MP-2-SR5):** Authenticating the payment request SMS uses the OTP security token for two-factor authentication. The charity's BPS authenticates the user by verifying the OTP included in the SMS, so if a phone is lost/stolen, an attacker cannot make a valid transaction without having the security token. The OTP is valid until it is received and processed by the BPS so network delays will not cause adverse effects. This should give some protection against spoofing attacks. The mobile phones handset's PIN protection will present a barrier to attackers who steal the phone. In a point-of-sale transaction, where a beneficiary is purchasing a product from a merchant, both parties can compare the TrNo received on confirmation messages before a purchased product is handed out. Social engineering may aim at obtaining privileged access to data at HQ/BPS, so security awareness training will be needed. However, an insider at the BPS/HQ/HQB is not able to transmit a transaction alone because of the use of multi-signature transactions/ smart contracts.

- **Availability (MP-2-SR6):** The donor platform and the BPS are attractive targets for DDoS attacks as they store keys and transaction data: see Table 5.12 for countermeasures. DDoS attacks against the blockchain are not viable due to its innate security and distributed nature.

- **Non-Repudiation (MP-2-SR7):** Every Bitcoin transaction uses digital signatures and confirmed transactions are recorded on the blockchain. The blockchain provides an immutable audit trail so a participant cannot deny their involvement.

SMS Security Issues:
The SMS system has well documented security issues, which are not addressed directly in this proposal. However, measures have been included which could deter would-be attackers. The use of the OTP means that replay attacks should fail, and the *AuthReq SMS* from the charity should alert users to potentially fraudulent transactions. Additional assurance comes from including both *TXID/RSKHash* and *TrNo* in confirmation SMS messages: these can be used to cross check with the Bitcoin/RSK blockchain and in a verbal comparison between Payer and Recipient respectively, to provide an extra level of assurance that a transaction is correct. This provides a higher level of security than other SMS Bitcoin schemes: e.g. in Coinapult SMS, the user confirms a transaction by sending an SMS containing a security code sent by the payment service in a previous SMS, which offers limited assurance that the transaction is genuine.

5.4.7 Use Case MP-2: Summary

The use case MP-2 first identified how a blockchain based solution could be employed with an SMS based m-payment system that uses the existing GSM network without requiring an Internet connection. It also uses an OTP based two-factor authentication method. The proposed SMS-based payment scheme was then evaluated for its security.

Table 5.13: M-Payment Use Cases vs Security Requirements

Security Requirement	**MP-1**	**MP-2**
Confidentiality	✓	Part[a]
Integrity	✓	Part[a]
Availability	✓	✓
Authentication	✓	Part[a]
Non-Repudiation	✓	✓

[a] SMS messages are not confidential/can be spoofed

5.5 Chapter Summary

This chapter discussed the security challenges of m-Payment applications, and then presented two use cases *MP-1:SCWS Branchless Banking* and *MP-2: Bitcoin SMS m-Payment* that offered secure solutions to these challenges. *MP-1* used the tamper-resistant security properties of the SCWS installed in a SIM with advanced capabilities

to provide secure mobile banking transactions of withdrawal, deposit and transfer of funds to a third party. *MP-2* used SMS messaging to interface with a distributed ledger system - the blockchain - to enable charities to offer humanitarian aid in an offline environment in the form of Bitcoin payments. The security of both use cases was assessed informally against previously defined security requirements and *MP-1* was also formally analysed using the protocol checking tool Scyther, and no attacks were found within bounds.

A summary of how well the security requirements were met for each use case is shown in Table 5.13.

5.6 Related Publications

Two publications resulted from the work described in this chapter:

1. S. Cobourne, K. Mayes, and K. Markantonakis, "Using the Smart Card Web Server in Secure Branchless Banking", in *International Conference on Network and System Security (NSS2013)*, Springer, 2013, pp. 250–263.[3]

2. D. Jayasinghe, S. Cobourne, K. Markantonakis, R.N. Akram, and K. Mayes, "Philanthropy On The Blockchain", in *11th WISTP International Conference on Information Security Theory and Practice (WISTP2017)*, 2017. [4]

Chapter 6

Authentication

Contents

Authentication is an enabler for secure services. This chapter firstly describes a proposal that uses the SCWS as a means of secure, local Single Sign-On (SSO) in a disconnected environment. Secondly, this chapter discusses the potential use of gesture recognition as a dynamic biometric authentication method on a mobile device.

6.1 Authentication Use Cases

Authentication is a vital enabler for secure services, and is usually based on one (or more) of the three authentication factors: *knowledge* - "something you know"; *ownership* - "something you possess"; and *inherence* - "something you are/do". Many of the security issues identified in this research can be attributed to weak authentication of an individual accessing a service.

The authentication use cases presented in this chapter utilise authentication factors in different ways. *Auth-1: Offline SCWS Single Sign-On* uses the tamper resistant web functionality of the SCWS installed on a SIM, with its many security advantages, to enhance authentication in offline environments. This provides two-factor authentication (2FA) through "something you have" (the SCWS) and "something you know" (an SCWS PIN).

Gesture recognition can provide one-step two-factor dynamic biometric authentication i.e. "something you know" (the gesture) with "something you are" (physical biometry). The advantage of dynamic biometry is that a biometric can be easily changed if compromised. Use case *Auth-2: Gesture Recognition Biometric* is concerned with the assessing the feasibility of utilising a mobile device equipped with a 3D depth camera to use gesture recognition as a dynamic biometric authentication method. Preliminary experiments are presented which use different 3D depth cameras (the Kinect and Leap Motion devices) for biometric data capture.

6.2 Authentication Security Requirements

Basic security requirements are shown in Table 6.1. The most important security requirements for an authentication system are Auth-SR3/Auth-SR4, as they are concerned with accepting correct users and denying access to imposters.

The next section now describes the first authentication use case, *Auth-1: Offline SCWS Single Sign-On*, where an SCWS on a phone SIM communicates with a second SCWS installed as part of a security module to provide authentication in an offline environment, through a Single Sign-On process using standardised web protocols.

6.3 Auth-1: Offline SCWS Single Sign-On

6.3.1 Single Sign-On (SSO) - Background Information

Single Sign-On schemes allow a user to authenticate online to a single central server, which then handles the user's authentication to other participating servers.

Table 6.1: Authentication - General Security Requirements

	Confidentiality
Auth-SR1	Sensitive information should not be disclosed to unauthorised parties, whether during processing, in transit, or at rest
	Integrity
Auth-SR2	Information must not be tampered with by unauthorised parties during processing, in transit or at rest, and a system must perform its tasks without unauthorised manipulation
	Authentication
Auth-SR3 Auth-SR4	Only genuine users can be authenticated successfully. No imposters can be authenticated successfully.
	Availability
Auth-SR5	Users must not be prevented from using the system: i.e. the service is not denied to authorised entities, for instance through distributed denial of service (DDoS) attacks.
	Non-repudiation
Auth-SR6	None of the participants can subsequently deny their actions took place

An example of an SSO scheme is the Kerberos protocol, (see Figure 6.1) where a "ticket" is obtained from a Kerberos Key Distribution Centre (KDC) and used as a trusted credential to authenticate to other services within the network.

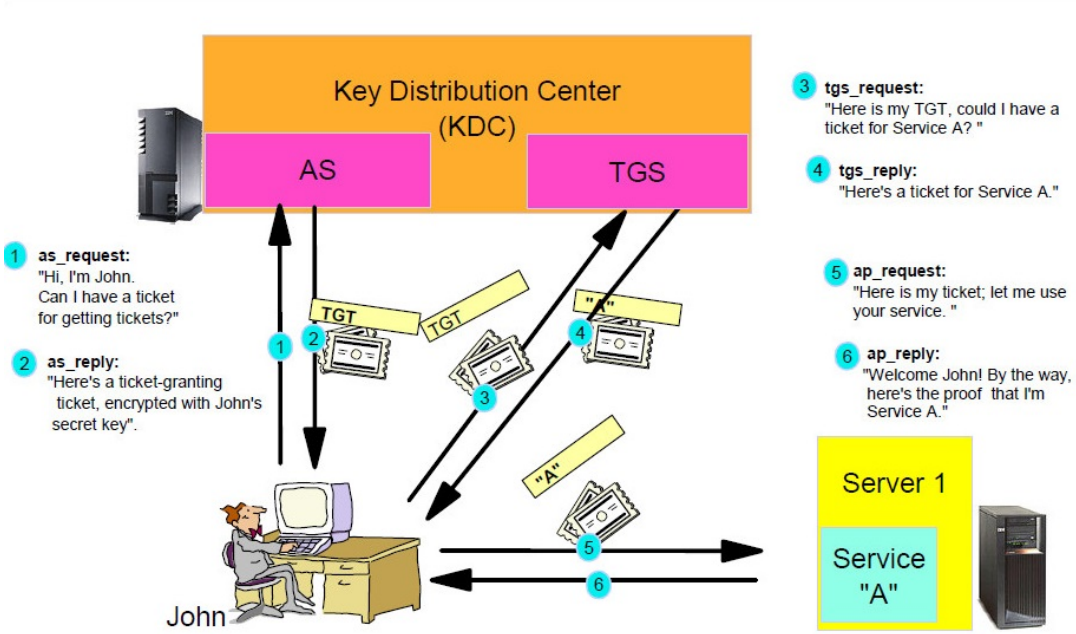

Figure 6.1: Kerberos Single Sign-On - Source: http://www.ibm.com/developerworks/

The next section describes how the SCWS can be used in an offline SSO solution. A full description of this proposal is included in the paper published as a result of this research [6].

6.3.2 Using the SCWS for SSO

The SCWS SSO protocol proposed in this chapter uses authentication tickets (tokens) like Kerberos. By installing an authentication application on an SCWS, it provides a decentralised mechanism, where users authenticate locally to their phones/readers. The authentication server is effectively within the SIM-SCWS of the user's mobile handset, personal to the user. So there is no centralised, single point of failure/ attack target for DDoS attacks. Authentication can also be performed in offline environments, using a variety of near-field channels to communicate.

For the proposal in this use case, an SCWS installed on a JavaCard v3.0 Connected Edition chip that supports a TCIP/IP protocol stack [235] is used to form a security module (referred to as MOD-SCWS) that will communicate with a corresponding application installed (more conventionally) on a SIM (referred to as SIM-SCWS). The MOD-SCWS needs to be installed within an electronic assembly that may be too expensive to use with cheap items (e.g in the Internet of Things), but can be integrated into high value equipment. The SIM-SCWS and MOD-SCWS can then be used to provide a local SSO scheme using standardised communication protocols like TLS [274].

The user authenticates themselves to the SIM-SCWS, with credentials previously stored inside the SIM card. This generates an authentication token (cf. "ticket" in Kerberos) to be transferred to the MOD-SCWS over a range of near-field communication routes, in order to gain access to sensitive information stored on the MOD-SCWS. Communication from the SIM-SCWS to the phone browser can be via ISO 7816, USB1.x and USB2.0 [275]: communication from SIM-SCWS to MOD-SCWS can be via Bluetooth Low Energy [276] or WiFi Direct [277]. Once the SIM-SCWS and the MOD-SCWS mutually authenticate, the MOD-SCWS retrieves the token stored in the SIM card and the user is authenticated if the token is valid and fresh.

The entities in the proposal are described in Table 6.2, and the relationship between them is shown in Figure 6.2. The necessary trust relationship between all the entities in the system can be achieved by a conventional X.509 certificate approach [278].

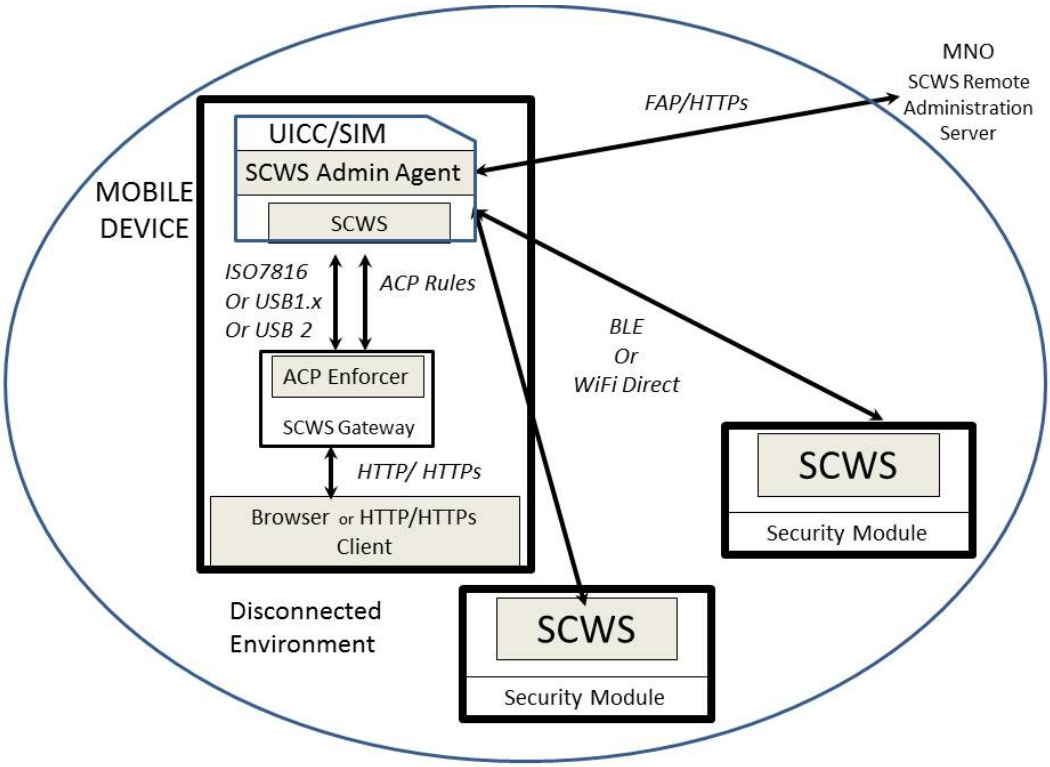

Figure 6.2: Auth-1: SCWS Offline SSO - Entity Diagram

6.3.3 Auth-1: Offline SCWS SSO Protocol

The SIM-SCWS should authenticate the user[1], the MOD-SCWS security module should authenticate the SIM-SCWS and vice versa.

The user PIN/password allows the user to authenticate to the SIM-SCWS, which creates an authentication token that is passed to the MOD-SCWS to gain access for a specific duration of time. A MOD-SCWS accepts the local authentication result, as it trusts the tamper-resistant SIM-SCWS. The authentication token is the concatenation of items shown in Table 6.4, i.e. $(L||T||I)$. The MOD-SCWS can communicate the requested data locally when tapped by an authenticated user/phone, without the need for online capability.

There are two phases in the proposed SCWS Offline SSO scheme: *Phase 1* - User Authentication/ Token Generation and *Phase 2* - Token Authentication. These phases can be seen in Figure 6.3.

[1]It is possible to have mutual authentication between the SIM-SCWS and the users browser through the exchange of certificates

Table 6.2: Auth-1: SCWS Offline SSO - Entities

Entity	**Description**
User	The user has a mobile device with an SCWS installed (SIM-SCWS): they communicate with both the SIM-SCWS and the MOD-SCWS over HTTPs via the phone browser
SP	Service Provider: the SP has a business relationship with the MNO in order to install their applications on the SIM-SCWS and MOD-SCWS.
MNO	Mobile Network Operator: the MNO owns and operates the SIM/SCWS ecosystem, and manages the administrative protocols that update content on the SCWS
RAS	Remote Administrative Server: the RAS is defined as a trusted entity in the OMA SCWS specification [15].
MOD-SCWS	A security module that can run the authentication application: it consists of an evaluated smart card chip supporting JavaCard v3.0 Connected Edition functionality, attached to an electronic assembly that can provide accurate time from an external timer.
SIM-SCWS	A Subscriber Identity Module (SIM) with an SCWS installed, which can run the authentication application

Table 6.3: Auth-1: SCWS Offline SSO - Assumptions

	Description
Auth1-A1	The MNO is trusted and allows access to the SIM card (for applets that will be used by the SCWS to run inside the SIM-SCWS).
Auth1-A2	There is a secure procedure for the development, deployment and revocation of certificates and key pairs on the various SCWS installations.
Auth1-A3	MOD-SCWS are installed on items that have a significant value, not tags or other low-power sensors
Auth1-A4	The MOD-SCWS receives accurate external time from an external timer found on the electronic assembly.
Auth1-A5	Each user has a single PIN/password that allows the user to authenticate to the SIM-SCWS, chosen, stored and updated in accordance with security best practice e.g. [242]

Table 6.4: Auth-1: SCWS Offline SSO - Protocol Notation

	Description
L	The level of access granted by the authentication token e.g. administrative or standard access
T	An expiration date/time
I	A unique SIM card identifier, for example the Integrated Circuit Card Identifier (ICCID) [279]

Phase 1 - User Authentication/ Token Generation: The user authenticates themselves to the SIM-SCWS using a PIN/password input to the phone browser: this is communicated to the SIM-SCWS via ISO 7816/ USB1.x or USB2.0. If successful, an authentication token is generated, signed and stored on the SIM-SCWS. The signature is created by the private key SK_{SIM} that is part of a key pair installed in the SIM-SCWS.

Phase 2 - Token Authentication: The user initiates a wireless connection between the SIM-SCWS and the MOD-SCWS: the two entities negotiate a TLS connection over local communication channels. The signed authentication token is then transferred to the MOD-SCWS. The MOD-SCWS checks the validity of the token, and verifies the signature of the token using the public key PK_{SIM} of the SIM-SCWS that is sent during the TLS handshake. The MOD-SCWS then logs the request and if the token has passed the verification check relevant information is sent to the phone browser over an HTTPs connection: if the check fails, then the token and the signature are discarded and the user is notified.

If a token is still valid, Phase 1 is not necessary so the user does not need to provide a PIN/Password every time, only once the token stored inside the SIM-SCWS has expired. The trust relationship between the SIM-SCWS and the MOD-SCWS means

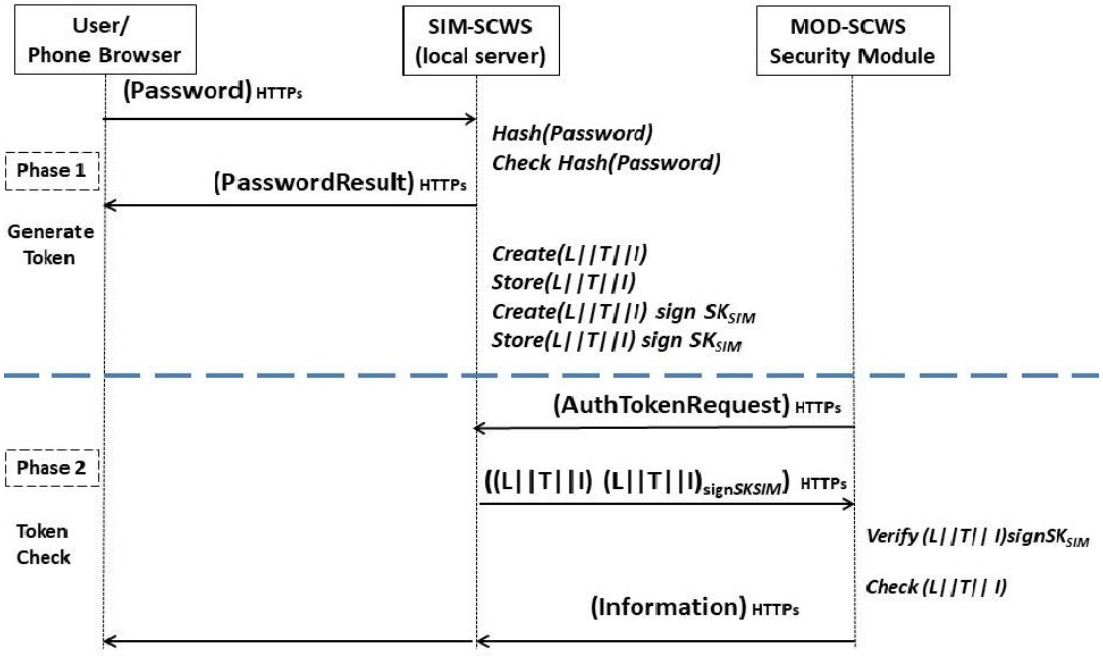

Figure 6.3: Auth-1: SCWS Offline SSO - Protocol

that the latter will always accept a valid SIM-SCWS token.

Submitting the token to the MOD-SCWS during Phase 2 of the protocol, ensures the freshest available token is always checked: authentication criteria may have changed between authentication attempts, as the user may have re-authenticated to the SIM-SCWS in the intervening period.

Performance timing estimates were done, for both processing and communication.

Total processing time was estimated as follows:

- **Phase 1** - User Authentication and Token Generation - **216.40 ms**
- **Phase 2** - Object Authentication and Access - **143.60ms**

Table 6.5 gives the estimated overall communication time needed for the whole protocol using a variety of communication combinations.

Table 6.5: Total Offline SSO Communication times (in ms)

SIM-SCWS to Browser	SIM-SCWS to MOD-SCWS	SIM-SCWS to MOD-SCWS
	Bluetooth LE	WiFi Direct
ISO 7816	2327.04	1568.69
USB1.x	1129.82	371.47
USB2.0	1121.27	362.92

From these estimates, it can be seen that the communication speed dominates, especially when the combination is for ISO 7816 along with Bluetooth LE. The full details of the method employed and the calculations can be seen in the paper published as a result of this work [6].

The next section assesses the security of use case *Auth-1* in comparison to the authentication security requirements shown in Table 6.1.

6.3.4 Security Analysis

The security properties of the SCWS installed on the tamper-resistant SIM are described in more detail later in this thesis, in Chapter 8. Defences that address authentication security requirements from Table 6.1 are as follows:

- **Confidentiality (Auth-SR1):** No sensitive data held on the MOD-SCWS is made available without authentication and all data transmissions are protected by HTTPs: this also applies to the local server SIM-SCWS. A major privacy advantage of the proposal is that the user's Password/PIN never leaves the local mobile device as it is checked locally by the SIM-SCWS.

- **Integrity (Auth-SR2):** The MOD-SCWS and local server SIM-SCWS will both protect the integrity of sensitive stored data; tamper resistance of the chip should mean it will function as intended, even if attacked. All involved entities communicate over HTTPs channels to protect integrity of data.
- **Authentication (Auth-SR3/4):** The MOD-SCWS checks the validity of the authentication token, and verifies the digital signature created by the SIM-SCWS. The authentication token will be unusable with any other SIM, as it contains a unique SIM identifier such as the ICCID. An attacker would also need the correct X.509 certificate in their malicious SIM card. The SIM-SCWS is accessed using a PIN.
- **Availability (Auth-SR5):** As the proposal avoids a single centralised SSO server by distributing the authentication process over a number of trusted local servers (SIM-SCWSs), there is no single point of failure. DDoS attacks cannot easily take place because an attacker has to infiltrate a large number of mobile devices/ MOD-SCWSs and this is not scalable. Denial of Service attacks on a single device may be possible, but the attacker would need to be in possession of the SIM-SCWS/MOD-SCWS for this to succeed.
- **Non Repudiation (Auth-SR6)**: the MOD-SCWS verifies the token's digital signature using the public key of the SIM-SCWS, and access requests are logged by the MOD-SCWS. This provides non-repudiation.

In the case of a lost or stolen mobile device, management procedures should ensure the authentication token is erased from the local server SIM-SCWS. Any non-authorised entity that tries to use it to access a MOD-SCWS will have to re-authenticate to create a new token, but will be presented with a barrier as they will not know the correct PIN/password.

Mutual authentication can be achieved if both SIM-SCWS and MOD-SCWSs have certificates. However the installation and maintenance of these certificates may be labour-intensive. The attack resistance of the chips could permit a longer period of certificate validity, depending on the value of the information stored on the chips.

6.3.5 Formal Security Analysis

The Scyther protocol verification tool was used to formally analyse the SCWS SSO protocol, and no attacks were found within bounds. The Scyther tool and its verification processes are fully described in Appendix B: the SSO Scyther Script and the verification results obtained are shown in Appendix Section B.4.

6.3.6 Use Case Auth-1: Summary

An SSO solution in a disconnected environment is possible using an SCWS installed both in an attack-resistant SIM card and a smart card chip (MOD-SCWS) embedded within an electronic assembly. The SIM-SCWS and MOD-SCWS communicate over a wireless communication channel: local authentication on the mobile device SIM-SCWS produces a security token that is accepted by the MOD-SCWS. Distributing trusted SSO authentication servers to many tamper-resistant SIM cards avoids a single centralised point of failure/ attack target for DDoS exploits. A password/PIN is used to authenticate the user locally on their mobile device, but attacks are not scalable as capturing this credential will not work without the corresponding SIM card.

The next section introduces use case *Auth-2: Gesture Recognition Biometric* that investigates the potential for using a mobile device to authenticate using gesture recognition as a dynamic biometric.

6.4 Auth-2: Gesture Recognition Biometric

6.4.1 Gesture Recognition Biometrics - Background Information

Gesture recognition could be used in a dynamic biometric system, as it can combine inherent physical information about the user with a knowledge factor (i.e. the gesture). Using gestures to authenticate benefits from a natural style which is particularly suited for non-technical users or those who are familiar with gaming or VWs. If gesture recognition could provide equivalent security to a 4-digit PIN then this would be a welcome additional authentication technique.

As with other biometric systems, gesture recognition needs both a sensor to capture the raw data from the gesture, and an analysis method to interpret the captured information. Matching decisions are based on a threshold of acceptance θ. This is used to decide how close the biometric input should be to a stored biometric template for it to be considered a match: θ can be varied for different operating environments/ security levels required. The accuracy of a biometric system can be measured using several means, for example using the False Positive Rate (FPR) which is when an imposter is able to be authenticated, or the True Positive Rate (TPR), which is when a genuine individual is authenticated correctly. The Equal Error Rate (EER) can also be used to compare the accuracy of different biometric systems. The EER occurs when FPR is equal to the False Negative Rate (FNR) (defined as FNR = 1-TPR) for a fixed threshold θ. A lower EER indicates that the system is more accurate.

Feature extraction identifies and extracts appropriate information from the cap-

tured data. Different analysis techniques can be used, such as Hidden Markov Models (HMM) [280]; Artificial Neural Networks (ANN) [281]; and Dynamic Time Warping (DTW) [282]. This use case focuses on the use of DTW as the analytic system, as it has the advantage that it needs less training samples than other classifiers such as ANN or HMM.

The next section discusses which mobile device sensors could be used for dynamic biometric data capture.

6.4.2 Mobile Device Sensors for Gesture Recognition

Accelerometers

The 3D accelerometer on a mobile device can be used to capture biometric data for analysis with the DTW algorithm, as seen in the work of [283]. Accelerometers are not ideal sensors for capturing gestures, as by recording only acceleration, useful information (such as speed and position) are lost, and they are sensitive to tilting of the user's hand when capturing gesture data. Further limitations occur because the device can only track one hand. Section 6.4.3 describes a preliminary experiment using the DTW algorithm with a mobile phone accelerometer as capture device.

A vision based sensor, such as a depth camera, can record gestures in 3D and would allow tracking of both hands.

Depth Cameras

There is a move within the smart phone industry towards equipping mobile devices with 3D depth cameras [284] to provide consumers with more feature-rich experiences such as augmented reality applications. There were announcements at the International Consumer Electronic Show (CES) in both 2016 and 2017 about phones with 3D depth cameras. In 2016, Google revealed that Project Tango [285] would add computer vision, depth sensing, and motion tracking technology to consumer mobile devices that year, to include full 3D video and time-of-flight cameras as popularised in the Xbox Kinect device [16] (illustrated in Figure 6.4). At CES 2017 the Asus Zenfone AR [286] was announced as a high-end Tango device. There are now several handsets available which are suitable for 3D depth camera applications [287]: for example, it was proposed to release a smart phone with integrated Intel RealSense 3D camera for $399 in 2016 [288]. As at June 2017, the Lenovo Phab2 Pro [289] was on sale at $499.99: it uses the world's smallest 3D camera [290] and is also compatible with Project Tango. A summary of the specifications and capabilities of these devices can be seen online [291]. The company behind the Leap Motion depth camera device [292] announced at the end of 2016 that

Figure 6.4: The Kinect Device [16]

they plan to install Leap Motion Virtual Reality technology into the same type of processors that are currently used in smart phones, thus opening up 3D tracking to battery-powered virtual and augmented reality devices [293].

So, having a 3D depth camera installed on a mobile device is no longer in the realm of science fiction: the possibilities this gives for gesture recognition based biometric authentication will now be explored through several preliminary experiments.

6.4.3 Auth-2: Preliminary Experiments

Experiments were done using a mobile phone accelerometer on a FxOS handset, the Alcatel Flame [79], and two different depth cameras, the Kinect and the Leap Motion devices. The Kinect is a structured-light 3D scanner that records video at 30 frames/second [294]. It captures and tracks a skeleton composed of 20 points in 3D: these represent the position of the head, neck, spine, centre hip, left and right side joints,

Figure 6.5: The Leap Motion - by SkywalkerPL, Attribution 4.0 International (CC BY 4.0)

hand, wrist, elbow, shoulder, hip, knee, ankle and foot. The Leap Motion device is an optical tracking system based on stereo vision that uses three infrared (IR) emitters that generate patternless IR light and two IR cameras (illustrated in Figure 6.5).

Using Accelerometer on FxOS Handset

As explained in Chapter 2, the Mozilla Firefox OS (FxOS) [78] was designed to work with very low specification handsets and limited data connectivity. Mozilla discontinued all work on the FxOS operating system in September 2016, but before that was announced, a proof-of-concept DTW gesture recognition biometric application was installed on an FxOS phone [79], using the device's accelerometer as a sensor. This application was written in Javascript, as FxOS is a web-based platform: sample screen shots can be seen in Figures 6.6 and 6.7. However, preliminary testing showed that the application was very sensitive to initial hand position, and meaningful results were hard to obtain.

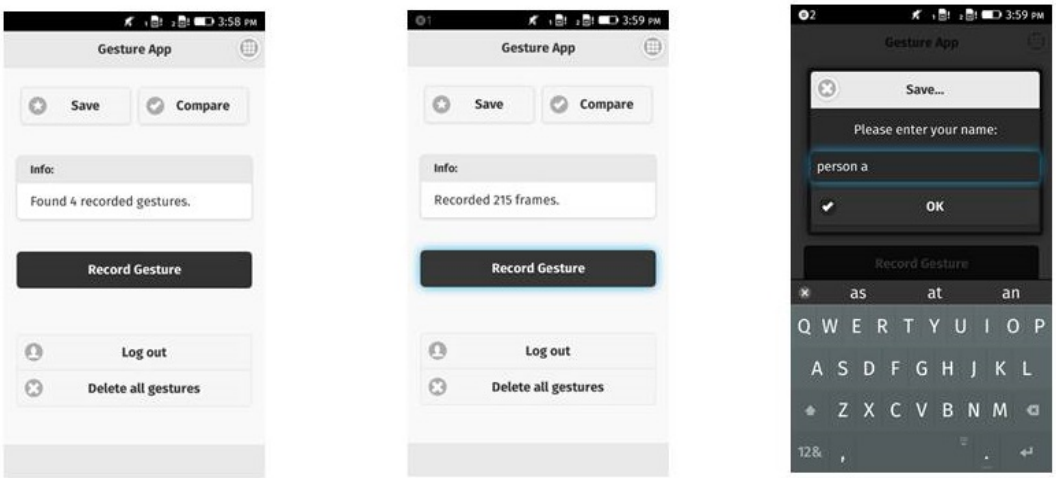

Figure 6.6: Auth-2: Accelerometer Record/ Save Gesture (FxOS Screenshots)

Using 3D Depth Cameras

As suitable mobile handsets with 3D depth cameras were not commercially available when this research work was done, preliminary experiments on the security, accuracy and robustness of gesture biometrics were done using off-the-shelf depth cameras i.e. a Kinect device as a sensor for large upper body gestures, and a Leap Motion device for smaller hand gestures. The DTW algorithm was used in all experiments.

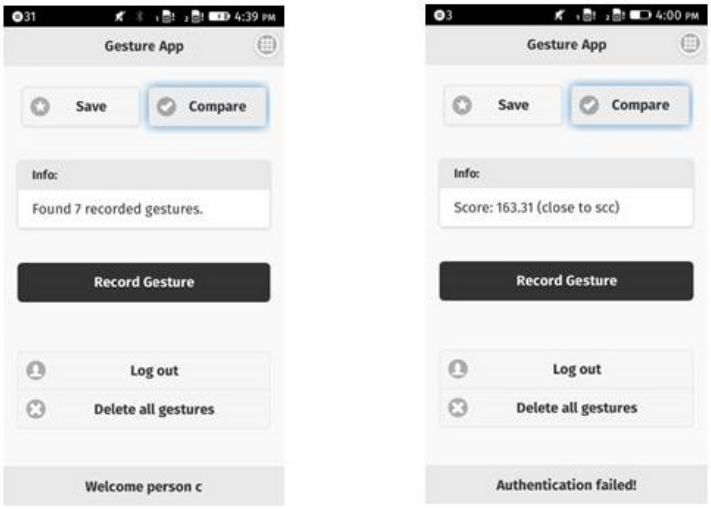

Figure 6.7: Auth-2: Accelerometer Authentication (FxOS Screenshots)

Using the Kinect: There has been previous work using depth cameras as biometric capture devices with DTW as the analysing algorithm: for example, in the work of Aumi et al. [295]. Other research has used the Kinect depth camera as sensor. Wu et al. [296] [297] used all 20 skeleton tracking points available on the Kinect device to give a True Acceptance Rate (TAR) of 98.11% for 1.89% of FAR. Tian et al. also used the Kinect with the DTW algorithm for analysis and recognition of gestures designing a 3D signatures [298], giving a TAR of 99% for 1% FAR and 3% against attacks.

In this preliminary experiment the Kinect was used to record large upper body gestures in an authentication experiment, using six of the available skeleton tracking points to study gestures made by arm and hand movements: the tracking points used were both hands, both elbows and both shoulders. The aim of the experiment was to assess the how well a gesture recognition biometric could withstand mimic attacks, as gestures are easily copied. Full details of the experimental method and analysis can be seen in the paper published as a result of the work [5], but the method and results are briefly summarised here.

The methodology for the experiment was to record gestures from one group of volunteers (the *reference group*) and ask a different group of volunteers (the *attacker group*) to attempt to copy them. There were 10 people in the reference group and 28 people in the attacker group.

Each member of the reference group was instructed to invent and perform a gesture that was an easily reproducible movement of their hands and arms, not too close to the body and less than five seconds in duration. These "reference gestures" were filmed

using a separate camera. Each reference group volunteer then repeated their gestures 20 times, which allowed the TAR/ FAR to be calculated. In the next stage of the experiment, each member of the attacker group was given 2 minutes to guess these reference gestures, without being told what gestures had been previously recorded. They were then shown a recording of a reference gesture, and they were asked to copy the gesture as accurately as possible 10 times: this was repeated for each reference gesture. This allows the FPR/ TRR (True Rejection Rate) to be calculated. In total, the experiment recorded 200 genuine authentication attempts, 56 minutes of guessing gestures, and 2800 attempts at copying reference gestures.

The results can be summarised as follows: the experiment gave an Equal Error Rate (EER) of 2.8%, a TAR of 93% and a FAR of 1.7% when attackers had been shown the reference gestures previously, simulating a shoulder surfing scenario where they can see the gesture being performed. If the attackers had not seen the gesture beforehand, then no attacks succeeded. So taking the total allocated time for attacks of this nature gave the result that the likelihood of a successful attack was less than 1 in 11200, a better security level than that a 4-digit PIN (which has 10,000 possible combinations).
Using the Leap Motion: In this preliminary experiment, small hand gestures were captured by the Leap Motion device by recording the (x, y, z) positions of the palm centre, all five fingertips and all five finger roots (i.e. eleven elements (E) for each frame). Leave One Out Cross Validation analysis was then done on all 11 points captured, as well as just the palm centre data points. Using palm centres removes hand geometry information, so gives similar data to an accelerometer sensor. This experiment resulted in 90,000 attacks and 10,000 attempts at authentication by genuine users. Full details of the experimental method and analysis can be seen in the paper published as a result of the work [7].

Palm centre gesture recognition was a less effective biometric than the full hand gesture as it has less input data. For a full hand gesture, an attacker mimicking a known gesture had 11.88% likelihood of a successful attack, whilst a genuine user had a 88.12% chance to be correctly authenticated: the equivalent figures for the palm gesture were TAR 74.96% and FAR 25.04%.

These experiments were used to test the performance of gesture recognition biometrics implemented in a "match-on-card" application installed both on a smart card and as an HCE application on an Android mobile. The full results can be seen in the published paper resulting from this work [5]. It was found that the overall processing time on the smart card was too long to be practicable, even using an optimised method of RAM management when performing the DTW calculations: HCE results were more promising, but would need extra security measures to be adopted as HCE does not

offer tamper-resistance.

6.4.4 Security Analysis

The investigation of dynamic biometrics on mobile phones has been primarily concerned with security requirements *Auth-SR3* and *Auth-SR4* in Table 6.1: the other security requirements relate more to whole systems.

The results obtained show that security levels better that the use of a 4-digit PIN can be obtained when using the Kinect for large upper-body gestures. The initial Leap Motion results were a little disappointing, however, and for that reason *Auth-SR3* and *Auth-SR4* for dynamic biometrics are assessed as "Partially Met" because accuracy of authentication is not consistent over different sensors. (The Kinect results would be considered to meet the security requirements). Studies based on the Kinect get better results than Leap Motion or short range depth sensors, which may imply that the more parts of the body that are used for the authentication, the better it is for correctly accepting the genuine user and the rejection of potential attackers. An attacker may be able to copy a gesture exactly after practising it several times.

In addition to making the biometric changeable, dynamic biometric schemes offer another security benefit. Improvements in unlinkability occur because the same physical characteristic can be used at different verifiers with different secret knowledge.

Table 6.6: Authentication Use Cases vs Security Requirements

Security Requirement	Auth-1	Auth-2
Confidentiality	✓	N/A
Integrity	✓	N/A
Authentication	✓	Partial[a]
Availability	✓	N/A
Non-Repudiation	✓	N/A

[a] The Kinect experiment fully meets these requirements: the Leap Motion is less accurate

6.4.5 Use Case Auth-2: Summary

Auth-2 investigated the potential for creating a gesture recognition dynamic biometric by using the DTW algorithm for analysis, with accelerometer and depth cameras as biometric capture devices. A proof of concept implementation of an accelerometer based system was installed on a FxOS phone, and preliminary experiments using 3D depth cameras (the Kinect and Leap Motion devices) were done. Security levels better than the use of a 4-digit PIN are possible when using the Kinect for large upper-

body gestures. In addition to providing a changeable biometric, dynamic biometry offers unlinkability improvements because a particular physical characteristic (inherent factor) can be used at different locations with a different knowledge factor (i.e. the gesture itself). In future, mobile devices will potentially be equipped with 3D depth cameras to provide users with a more feature-rich experience. The results obtained from the preliminary experiments indicate that 3D cameras could potentially be used for gesture recognition dynamic biometric authentication on mobile devices in future.

6.5 Chapter Summary

This chapter presented two use cases *Auth-1: Offline SCWS Single Sign-On* and *Auth-2: Gesture Recognition Biometric*. *Auth-1* used the tamper-resistant security properties of the SCWS installed in a SIM (SIM-SCWS) and a security module with a smart card chip (MOD-SCWS) embedded within an electronic assembly to provide an SSO solution for disconnected environments. Local authentication on the mobile device SIM-SCWS produces a security token that is accepted by the MOD-SCWS. This distributed authentication approach avoids a single centralised SSO server, and attacks against an individual SCWS are not scalable as they require possession of the SIM-SCWS or MOD-SCWS to succeed. The security of use case *Auth-1* was assessed informally against previously defined security requirements and formally using protocol checking tool Scyther, and no attacks were found within bounds.

Use case *Auth-2* investigated the potential for using depth cameras on smart phones as sensors for dynamic biometric authentication, using the DTW algorithm to analyse captured data. As at the time of writing, depth camera phones were not commercially available, preliminary experiments were done using the Kinect and Leap Motion devices to assess the accuracy and practicality of the approach. Results showed that small hand gestures used with the Leap Motion were less accurate than upper body gestures captured by the Kinect.

A summary of how well the security requirements were met for each use case is shown in Table 6.6.

6.6 Related Publications

Four publications resulted from the work described in this chapter:

1. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Authentication Based on a Changeable Biometric using Gesture Recognition with the Kinect™",

in *2015 International Conference on Biometrics (ICB2015)*, IEEE 2015 pp. 38–45. [5]

2. L. Kyrillidis, S. Cobourne, K. Mayes, and K. Markantonakis, "A Smart Card Web Server in the Web of Things," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys 2016)*, Springer, 2016, pp. 769-784 [6]

3. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Comparison of Dynamic Biometric Security Characteristics against other Biometrics", in *2017 IEEE International Conference on Communications (ICC2017)*. [7]

4. B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Gesture Recognition Implemented on a Personal Limited Device", in *8th International Conference on Information and Communication Systems (ICICS2017*) [8] (Nominated for Best Paper)

Chapter 7

Virtual World Applications

Contents

This chapter proposes an enhanced login procedure suitable for VWs that harnesses the excellent security properties of the SCWS, along with OTP processes and geolocation. This is followed by a proposal for a remote code voting solution for in-world VW voting, where sensitive operations are processed in a Trusted Secure Layer external to the VW infrastructure, and Vote Code Lists are sent via the mobile network as a second channel to a mobile phone application.

7.1 Virtual World Use Cases

Two of the chosen application areas will now be investigated in the VW environment: authentication and e-voting.

Weak authentication of users of VWs can lead to identity theft and other real world problems for users. This issue is investigated in use case *VW-1: SCWS Online VW Log-In* using an enhanced log-in procedure that harnesses the excellent security properties of the SCWS, along with geolocation to authenticate a RW user to the VW.

VWs are social environments: in 2008, ENISA found that 59% of VW users joined a community/guild [181]. In SL [172] groups are managed by a group administrator, who determines which avatars are eligible to join. Keeping in-world VW votes private is difficult because of constant monitoring of in-world activities, so elections are mostly held outside the VW environment, in forums/wikis (such as [299]), detracting from the immersive VW experience. Use case *VW-2: VW Voting* proposes an in-world voting solution where the most sensitive e-voting processes are done in a Trusted Secure Layer external to the VW infrastructure, in conjunction with Vote Code Lists sent to a mobile phone application.

Note: The M-Payment application area is not applicable to the VW environment as the payment infrastructure uses standard external RW e-commerce facilities such as PayPal, or in-world VW currency.

The next section now describes the first Virtual World use case, *VW-1: SCWS Online VW Log-In*.

7.2 VW-1: SCWS Online VW Log-In

Most VWs use a simple username/password as the user authentication mechanism at login. However this relies on static data for security, which may be captured via a range of security attacks, and exploited subsequently[1]. Strengthening VW authentication should improve confidence in the environment, allowing for the introduction of new services, such as banking, insurance, and in-world voting, which are currently held back by security concerns regarding the identity of the RW controllers of avatars.

[1]Some VWs have taken a different approach to identification, however: for example, Blizzard has taken extra security measures with regard to WOW authentication. They have an optional Battle.net Authenticator, which is either a physical token or an application on supported mobile devices, that can be used for two-factor authentication to protect against unauthorised account access [28]. Also, in addition to player-chosen nicknames (BattleTags), Blizzard have introduced a voluntary, optional level of identity called RealID into WOW [300], where Real ID friends are made aware of their WOW friends' real name, but no other personal information.

7.2.1 VW-1: Security Requirements

The security requirements that an online VW Login system should meet are shown in Table 7.1. At the successful conclusion of the protocol, there should be an assurance that if avatar X is in the Virtual World then its legitimate real world controller (user X) has correctly used two-factor authentication i.e. "something you have" (SCWS and phone) and "something you know" (SCWS PIN and OTP).

Table 7.1: VW-1: SCWS Online VW Log-In - Security Requirements

	Confidentiality
VW1-SR1	Sensitive information should not be disclosed to unauthorised parties, whether during processing, in transit, or at rest
	Integrity
VW1-SR2	Information must not be tampered with by unauthorised parties during processing, in transit or at rest, and a system must perform its tasks without unauthorised manipulation
	Authentication
VW1-SR3	Only authorised users and the trusted RAS can access/ update the SCWS.
VW1-SR4	Only authorised users can be logged in to the online service: the protocol must ensure that the user is not an impostor, and is a real world individual
	Availability
VW1-SR5	Users must not be prevented from using the system: i.e. the service is not denied to authorised entities, for instance through DDoS attacks.
	Non-repudiation
VW1-SR6	None of the participants can subsequently deny their actions took place

In the next section, an authentication solution is proposed that uses a SIM equipped with an SCWS, OTP processing and geolocation to strengthen authentication to the VW Login Server[2]. Full details of the system can be seen in the paper published as a result of this work [9]: a summary is now presented.

7.2.2 Using the SCWS for Online VW Log-in

The general principle behind the proposed protocol is that the user enters a VW username into the VW Client, and this triggers an OTP generating process at the VW back-end. A time-constrained OTP is produced using a nonce N and the user's PIN,

[2]This solution is presented using Online VW log-in as an example, but the protocol is equally applicable to other RW server applications.

Table 7.2: VW-1: SCWS Online VW Log-In - Entities

Entity	**Description**
User	The user has a mobile device with an SCWS installed (SIM-SCWS): the user communicates with the SIM-SCWS over HTTPs via the phone browser: the User is an individual who is registered with a VW
VW	Virtual World: has details of all user credentials. It is responsible for checking user login details and creating OTPs.
MNO	Mobile Network Operator: the MNO owns and operates the SIM/SCWS ecosystem, and manages the administrative protocols that update content on the SCWS
RAS	Remote Administrative Server: this is the intermediary between the VW Server and the mobile SCWS. The RAS can also determine the phone's location (as it is a part of the MNO) and inform the VWS/Login Server.
VWS	Virtual World Server: Provides back-end functionality for the VW, and has all the necessary information to keep the world operating for the user and also connect with the back-end process such as the database and the login server.
LS	Login Server: part of the Virtual World, this manages login details, authenticates VW avatars, creates the OTP and a nonce (N) and checks the OTP sent by the VWC.
SCWS	Smart Card Web Server: the user accesses the SCWS environment using a PIN. The SCWS uses a java applet for processing information securely.
VWC	Virtual World Client: this is the interface presented to the User, and is installed on the user PC. The VW Client provides only a graphical user interface, and is considered insecure.

and the nonce is then sent to the user's SCWS-enabled phone. The user enters a PIN and the OTP is re-created locally. The displayed OTP on the phone must be input to the VW Client on the PC, which forwards it to the VW Server for the authentication decision. (More information about OTPs can be seen in Appendix A.1.1.) Additionally, the VW Login Server checks the phone's location against the PC's location, using the cell towers of the mobile network/ the mobile's GPS adapter/ IP geolocation as required. If both platforms are in the same geographic area then the authentication may proceed.

The entities involved in the proposal are described in Table 7.2, and the relationship between entities is shown in Figure 7.1. Necessary protocol assumptions are shown in Table 7.3, and protocol message flow is shown in Figure 7.2.

There are five phases in this authentication proposal:

- **Phase 1:** The user makes a log-in request at the VW Client, entering their username or other credential.

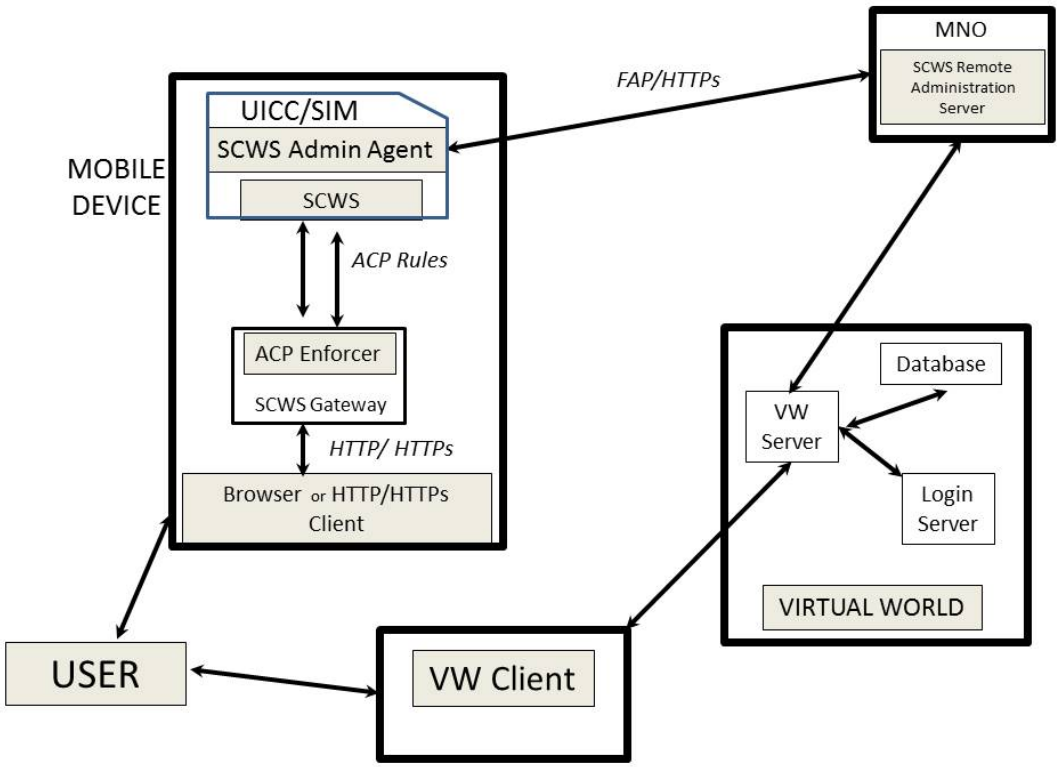

Figure 7.1: VW-1: SCWS Online Login - Entity Diagram

- **Phase 2:** The user's (previously supplied) PIN is used by the VW Login Server to create an OTP_{VWS}, along with a nonce N.
- **Phase 3:** The VW Server gets and stores the phone location, and sends N to the SCWS. The RAS obtains the phone's location and passes it back to the VW Login Server.
- **Phase 4:** The User enters their PIN to the SCWS, which uses it to create OTP_{SCWS}.
- **Phase 5:** The user enters the OTP_{SCWS} (displayed on the phone) into the VW Client. This is passed to the VW Login Server for an access decision based on whether the two OTPs match and the geolocation check is satisfactory.

A preliminary security analysis of the proposal will now be conducted.

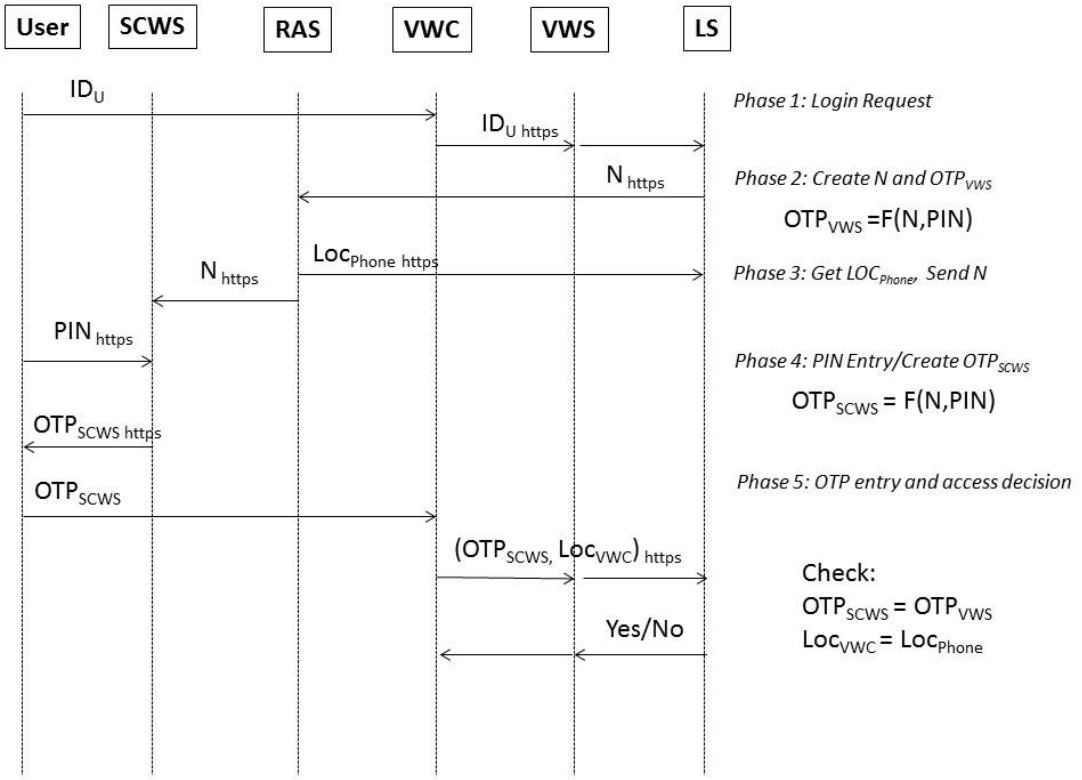

Figure 7.2: VW-1: SCWS Online VW Log-In - Authentication Protocol

7.2.3 Security Analysis

The security of the SCWS installed on the tamper-resistant SIM is described in more detail in Chapter 8. However, there are several potential attack points which need to be considered with regard to the stated security requirements (see Table 7.1, and these will be now be analysed[3].

Confidentiality (VW1-SR1): Attacks could be mounted as follows:

- *Attacks on the VWC:* The VW Client is the easiest target to attack, but in this proposed system, stealing a username/user ID is not enough to complete the login, as a SCWS/phone is used as a second factor. If there is a physical attack on the VW Client equipment i.e. the PC is stolen, the malicious user will not be able to login into the VW system as the real authentication secret is sent to the phone.

[3]There are generic attacks which are beyond the scope of this research: the protocol does not seek to address compromise of the VW Server or Login Server.

Table 7.3: VW-1: SCWS Online VW Log-In - Assumptions

	Description
VW-1-A1	The MNO has a business relationship with the VW developers, and has authorised their use of a Remote Administration Server to update the SCWS VW application and data.
VW-1-A2	The mobile phone has a one-to-one mapping to the user, i.e. only one registered user can use a particular SCWS.
VW-1-A3	A secure user registration procedure is in place: the user will set a PIN to access the SCWS, and must supply a mobile phone number to the VW and authorise its use so that the RAS will be able to download the Java applet/credentials onto the correct phone using techniques described in Chapter 3. Additionally, during registration, a VW certificate will be installed on the user's VW Client to offer mutual authentication.
VW-1-A4	The HTTPs channels between the RAS and SCWS, and between the VW Client and VW Server are considered secure.

- *Malware on PC:* If there is malware on the client PC (while still in possession of the genuine user) e.g. a keylogger which records the OTP as it is input by the user [301], it will not gain any advantage, as the short validity period of the OTP means that authentication will probably complete before the malicious entity can mount an attack.

- *Stolen Phone*: if both the phone and PC are stolen then the attacker would also have to know the SCWS PIN to retrieve the OTP information from the SCWS, or physically attack the SCWS/SIM card. Management procedures should ensure the authentication applet is disabled on the SCWS via the RAS if the phone is reported lost/stolen.

- *Malware on phone* - this would not be able to read credentials stored in the tamper-resistant SCWS, as access is restricted to authorised applications controlled by the ACP Enforcer

- *Privacy*: no sensitive data held on the SCWS is made available without authentication (via PIN) and all data transmissions are protected by HTTPs. The user's PIN never leaves the local mobile device as it is checked and processed locally by the SCWS.

Integrity: (VW1-SR2)

- *Data at rest:* the SCWS will protect the integrity of sensitive stored data; tamper resistance should mean it will function as intended, even if attacked.

- *Data in transit:* this data is protected by HTTPs at all stages: from phone browser to SCWS, RAS to SCWS and between the VW Client and VW Server.
- *Replay attacks:* Replay attacks do not work as an OTP is time-constrained and, by definition, only used once.

Authentication: (VW1-SR3/4)

- *User Authentication:* The user authenticates to the SCWS using a PIN - if the PIN is stolen it will be unusable with any other SIM, so the attacker has to be in possession of the phone also. The OTP entered into the VWC is dependent on credentials stored within the SCWS, which is tamper-resistant.
- *VWC Authentication:* There is mutual authentication between the VW Client and VW Server as the client is given a VW certificate during registration: malware would also have to steal or replicate this certificate for a successful attack.

Availability: (VW1-SR5)

- *DDoS Attacks*: The proposal distributes the authentication process to a trusted local servers i.e. the SCWS, so there is no single point of failure.
- *PC malware*: If PC malware is able to perform a DoS attack against the VWC, so that the user will not be able to submit the OTP, the inclusion of geolocation means that the IP address of the user must also be spoofed for it to succeed. Although this attack is possible [302], the attacker needs to simultaneously know the location of the user /phone, spoof the IP, and mount a DoS attack against the VWC, which add extra deterrents.

Non Repudiation: (VW1-SR6)

- *Recording Login Attempts*: The Login Server will keep a record of all login attempts to provide non-repudiation.

7.2.4 Use Case VW-1: Summary

Relatively weak user authentication procedures often employed by VWs that rely on static username/password combinations can lead to security issues such as identity theft. The use case VW-1 has presented a method for using a phone, equipped with an SCWS SIM, to enhance the log-in procedure of VWs by using OTPs and location based checks. The use of a static password is replaced by that of a dynamically created one (different at each login attempt) created on a separate personal device, using a tamper-resistant chip (the SIM), and connected via a different communications channel. A

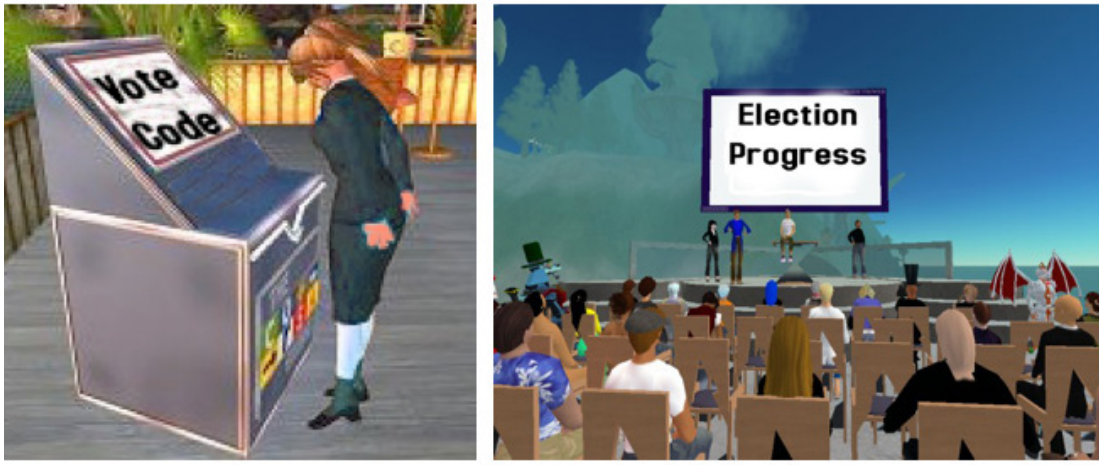

Figure 7.3: Virtual World Voting - Polling Booth Example

preliminary security analysis was done which indicated that the proposal had promising capabilities to prevent unauthorised access to VWs.

The next section describes the application area of remote e-voting applied to the VW environment, in use case *VW-2: VW Voting*.

7.3 VW-2: VW Voting

7.3.1 VW Voting - Background Information

VWs are social environments; group decisions can be made through voting in elections, but this is mostly done outside the VW environment, in forums/wikis (such as [299]). Some VW elections attract large numbers of voters e.g. in 2010, the VW Eve Online's "Council of Stellar Management (CSM)" election [299] had 39,433 votes from over 20 countries. In-world voting would increase the realism of the VW, especially if a VW plans to introduce revenue-sharing/political systems [303]. In-world voting has been seen in SL, through "voting stations" (also known as "voting booths", "voting boxes" and "vote boxes"), that were used to generate virtual money for avatars who had created appealing virtual objects/ buildings. Avatars would click on the voting station to cast their vote to indicate their approval for the object and the creator would be rewarded later. However, these were superseded and subsequent schemes have been discontinued following abuse [304].

Possible VW voting scenes are shown in Figure 7.3, visually mimicking normal RW voting.

Privacy concerns arise in VWs because avatars' communications and activities are

monitored by VW developers using sophisticated data mining and behaviour analysis to detect in-world cheating, through techniques similar to spyware [181]. Legal protection such as the EU Privacy Directive [182] only applies to natural persons i.e. the avatars' RW controllers.

7.3.2 VW-2: Security Requirements

As seen in Chapter 4, voting securely is demanding task, in any environment. Within a VW, however, extra challenges exist as both generic e-voting and VW-specific security issues must be addressed. e-Voting security requirements that must be met are shown again in Table 7.4. In VW voting, no RW data should be accessible to in-world entities, and a vote in the VW should not be linkable to an avatar or its RW user.

In this proposal, a code voting approach was selected as the most suitable solution to the secure platform problem for voting in VWs.

Table 7.4: VW-2: VW Voting - Security Requirements

	Confidentiality
VW2-SR1	Secrecy of the vote (*privacy*)
VW2-SR2	Vote cannot be traced back to a voter (*unlinkability*)
VW2-SR3	Voter can vote without external influence (*vote-buying/ coercion*)
	Integrity
VW2-SR4	Votes should not be tampered with (*recorded as cast*)
VW2-SR5	Votes should be included correctly in the final election result (*counted as recorded*)
	Authentication
VW2-SR6	Only eligible voters can vote (*democracy*)
VW2-SR7	Voters can vote only once (*democracy*)
	Availability
VW2-SR8	Voters must not be prevented from voting *(forced abstention/ denial of service*)

Code Voting

In code voting, a VA randomly generates voting authorisation codes, and assigns them to candidates in a Vote Code List (VCL). The VCL is sent to voters before the election, via a second secure channel such as the postal service. Figure 7.4 shows the functions of a typical code voting scheme, and Figure 7.5 shows example VCLs. It can be seen that VCLs have Vote Codes VC_X confirmation codes CC (one per VCL or one per candidate) and a (hard-to-guess) random VCL reference ID *(*ID_{VCL}*)*. To vote, the candidate's vote code is entered: a valid vote has the format (ID_{VCL},VC_X). As the

codes are random numbers, an attacker cannot tell who the vote is for without prior knowledge of the VCL. The confirmation code is sent to assure the voter that the vote has been received correctly by the system. It is desirable for an election to be verifiable by individual voters/third parties, to check votes have been recorded and counted correctly: publicly accessible Web Bulletin Boards (WBB) storing encrypted vote information can be used here. Example code voting schemes can be seen in [86, 18, 91, 92].

Code voting has a trust assumption that the secrecy and unlinkability of VCLs will be maintained, i.e. the security provided is not suitable for high-security elections but should be adequate for VW voting.

In this use case, a code voting system is located in a Trusted Secure Layer (TSL), external to the VW Server (VWS). This sends VCLs to the user's RW phone via the mobile network. This retains some RW elements (i.e a phone), but votes can be cast without leaving the VW environment. Knowledge fragmentation across several environments can then preserve the privacy of the avatar's vote.

7.3.3 VW-2: VW Voting Using a Trusted Secure Layer

The Trusted Secure Layer (TSL)

VW processes can be insecure: data necessary for displaying and controlling the user interface is transferred from the VW Client (VWC) to the VWS unencrypted [305], so man-in-the-middle attacks can tamper with this data and potentially send the VWS

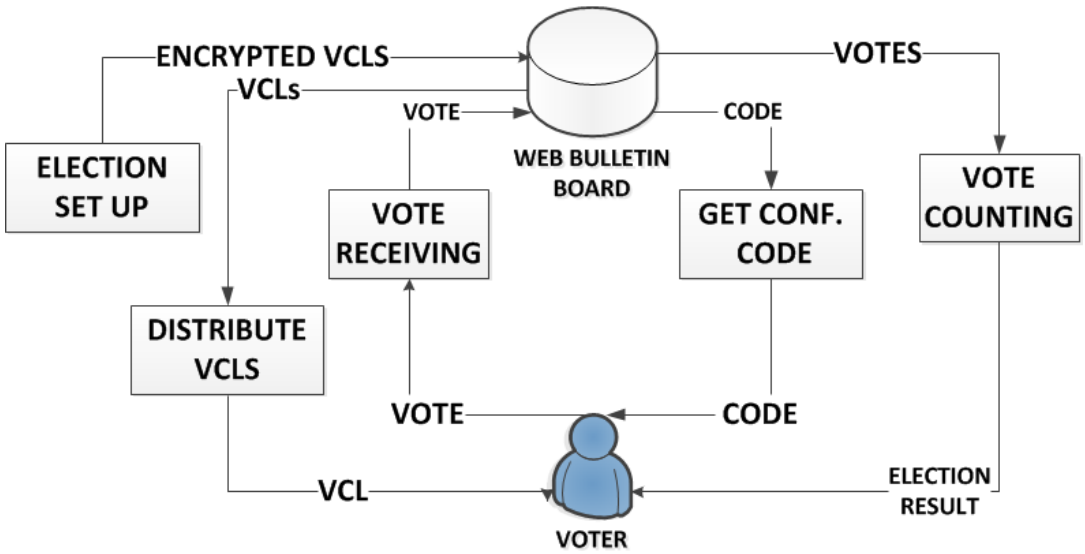

Figure 7.4: Code Voting Functions

Example A

NAME	VOTE CODE	CONFIRM CODE
CAND 1	143568	852369
CAND 2	578963	741235
CAND 3	753695	563254
CAND 4	448596	789541
VCL ID: 123456789012		

Example B

NAME	VOTE CODE
CAND 1	143568
CAND 2	578963
CAND 3	753695
CAND 4	448596
CONFIRM:	854697453
VCL ID:	13243546576

Figure 7.5: VCL Examples:(A) SureVote [17] (B) PGD [18]

logic into an undesired state. This use case proposes to locate sensitive e-voting processed in the TSL external to the VWS, inaccessible from the VWC. The TSL has only two connections (both full duplex), to the mobile phone network and the VWS. These are trusted choke points with well defined functionality and minimal data in transit across them, reducing the TSL attack surface. Filtration and categorisation of message/datagram packets is enforced, and only well-formed traffic is allowed. Also, white listing checks (i.e. connections come from trusted sources) are done and traffic from unknown sources is discarded. Data held within the TSL is encrypted using standardised algorithms and best practices e.g. [241]. Secure communication protocols (such as HTTPs) are used for all traffic into or out of the TSL: direct dedicated communication lines could be used between the VW and TSL. Additionally, vote codes can be sent from the TSL to the user's mobile phone, bypassing the insecure VW.

The proposed protocol spans four distinct zones of operation: the RW, inside the VW itself, the TSL external to the VW, and the VW Server. The entities necessary for the proposed voting protocol are described in Table 7.5: their relationship is shown in Figure 7.6. Assumptions made are shown in Table 7.6, and notation shown in Table 7.7. Entities in the TSL are fully trusted, others are not.

7.3.4 VW-2: VW Voting Protocol

There are four stages required in an e-voting solution: Registration/Activation; Election Setup; VCL Request; and Voting.

Table 7.5: VW-2: VW Voting - Entities

Entity	**Description**
TSL: e-Voting Server (e-VS)	This provides code voting functions as in Figure 7.4
TSL: TSL: Registrar (R)	This registers a user's RW mobile phone number and distributes VCLs
VW Server (VWS)	This communicates with avatars via the Polling Booth (PB) object in the VW. It has VW avatar/group/election databases, but does not store any sensitive voting credentials.
RW: User (U)	An individual who controls one or more avatar(s). Users need a VW Client on their PC, and a voting application on their mobile phone.
RW: Phone	A phone application which receives and displays VCLs sent from the Registrar over a secure mobile network communication channel.
VW: Avatar (A)	An avatar controlled by a RW user.
VW: Polling Booth (PB)	A VW object controlled by the VWS, a "dumb terminal", with no cryptographic capability.
VW: Group Administrator (GA)	An avatar authorised to create group elections, possibly with a registered phone.

The protocol message flow can be seen in Figure 7.7.

Registration/Activation is done once per avatar. A one-time activation code AC is used by the Registrar (R) in the TSL to link the user's phone ID_{PhU} with the hash of

Table 7.6: VW-2: VW Voting - Assumptions

	Assumption
VW2-A1	Avatar UID_As are uniquely identified by the VW Server: e.g. in SL, a user account has a main avatar and up to 5 extra "alts", each allocated a unique username (32 Hexadecimal Code) by the VW Server.
VW2-A2	The links between Vote Codes/VCL ID are not leaked
VW2-A3	The RW user can have many VW avatars
VW2-A4	Mobile network communication between the TSL/ phone is trusted
VW2-A5	All communication/ activities within the VW environment can be eavesdropped, intercepted, manipulated and replayed by attackers
VW2-A6	When information is input to the VW a CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) [306] should be used to ensure that the data source is not a scripted bot.
VW2-A7	There is no collusion between the TSL and any other entity.
VW2-A8	All messages will include appropriate freshness mechanisms.
VW2-A9	All RW credentials held in the TSL are protected cryptographically.

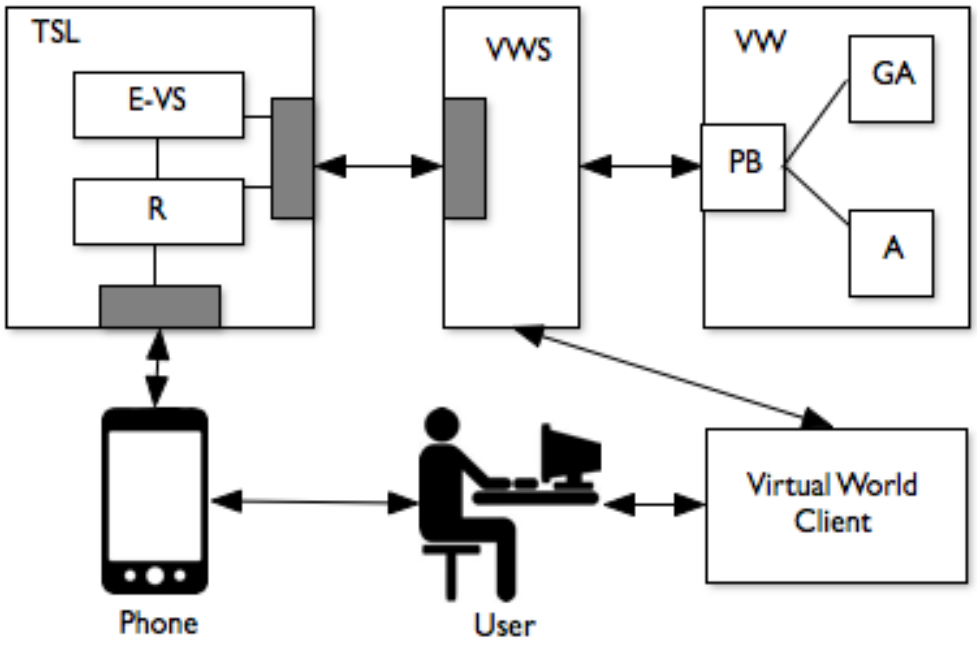

Figure 7.6: VW-2: VW Voting - Entity Diagram

the avatar ID H(UID_A): H(UID_A) is sent back to the user's phone.

Election Set Up is required for each election. In the VW a GA inputs group ID, election parameters EP and ballot data BC_1... BC_M at a PB. The VWS checks the input data and generates generates election reference REF_E and *Roll* i.e. the number of avatars eligible to vote in an election. It also initialises voting status flags for all eligible voter avatars to zero $status_{A1}$.. $status_{AX}$ flags (set to 0). The VWS forwards ballot details to the e-VS, which sets up the WBB and VCLs (enough for *Roll* voters plus auditing).

Request VCL is done when an avatar enters REF_E into a VW Polling Booth (PB). The VWS checks UID_A eligibility on election database, then creates/stores H(UID_A) The VWS sends REF_E, $H(UID_A)$ to R and sets $status_A$=1 (VCL requested) R requests and receives a VCL from e-VS and sends the VCL and H(UID_A) to Ph_U: R notes that H(UID_A) has been sent a VCL (but not which VCL). via $status_A$=2 (VCL sent).

Voting is done at a PB when the avatar enters REF_E. The VWS uses UID_A to check $status_A$=2 The ballot is displayed on the PB and the user's RW phone displays the

Table 7.7: VW-2: VW Voting - Protocol Notation

Notation	Description
A	Avatar (VW entity)
AC	Activation code
BC_M	Ballot Choice M
CC_M	Vote confirmation code for Ballot Choice M
VWS	VW Server(entity)
EP	Election Parameters e.g. name/ date/ time/ eligibility
e-VS	e-Voting Server (entity)
G	Group (of avatars) in VW
GA	Group Administrator (VW entity)
H(Z)	Hash of data Z using a standardised algorithm/best practices as specified in [307])
ID_X	Identity of entity X
No_X	Random Nonce generated by entity X
PB	Polling Booth (VW object)
Ph_X	Phone number of entity X
R	Registrar (TSL Entity)
REF_E	Election reference number
Roll	Number of avatars eligible to vote in an election
$status_A$	Avatar voting status flag, values =0,1,2,3,4
U	User (RW entity)
UID_X	Unique identifier for VW entity X
VC_M	Vote code for Ballot Choice M
VCL	Vote Code List (BC_1,VC_1,CC_1...BC_M,VC_M,CC_M ID_{VCL})
I→J:	Message sent from entity I to entity J

VCL. At the PB, the avatar inputs ID_{VCL} and VC_N for their ballot choice BC_N. The VWS sets $status_A$=3 (vote in progress) The VWS adds a nonce for freshness No_{VWS}, and sends the vote to the e-VS. The VWS knows which avatar entered which vote code, but not which ID_{VCL}s are valid, nor which candidate VC_N relates to. The e-VS obtains/recalculates the matching CC_N, posts the vote to the WBB to include it in the counting process (possibly after a short delay in case an observer in the VW links the vote posted to an avatar who has just voted). It should be noted, that the vote receiving process in the e-VS only knows valid ID_{VCL} values, not the contents of the VCL itself. The e-VS sends CC_N to the VWS with No_{VWS}+1. The VWS checks No_{VWS}+1, displays CC_N at the PB, then updates $status_A$=4 (voted) for the associated UID_A, and the RW user can check CC_N displayed on the PB received matches CC_N on the phone.

A preliminary security analysis of the proposal now follows.

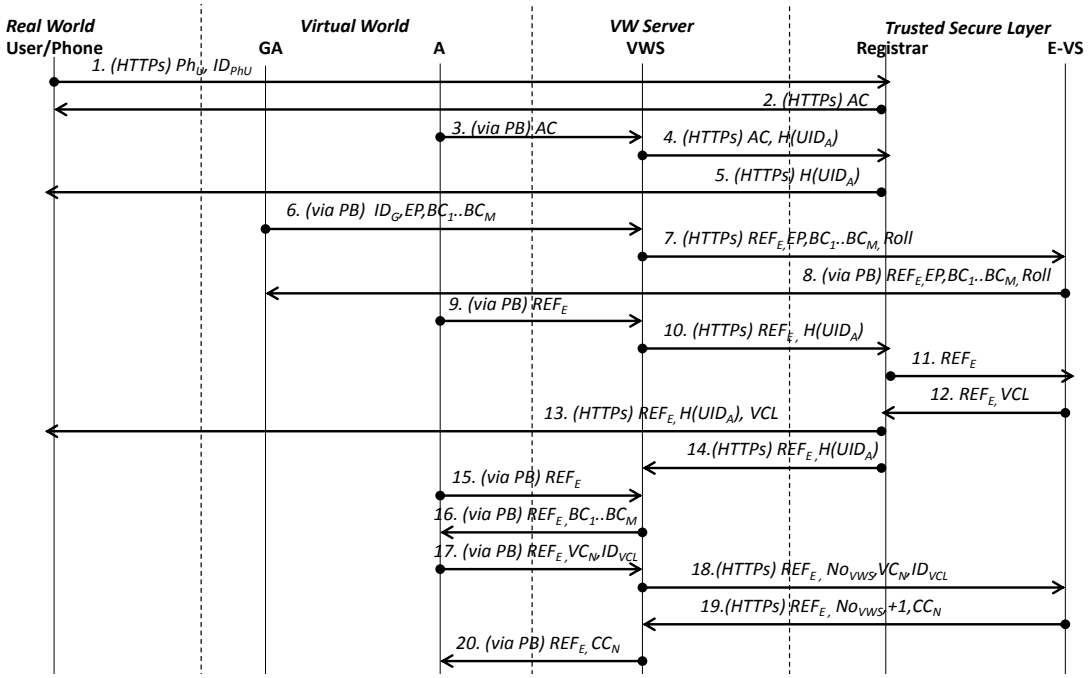

Figure 7.7: VW-2: VW Voting - Protocol

7.3.5 Security Analysis

The security of the proposal will now be analysed with respect to the security requirements shown in Table 7.4. (More details about code voting security can be seen in [18, 91]).

Confidentiality (VW2-SR1/SR2/SR3):

- **Privacy (VW2-SR1):** The basis of code voting security is that without prior knowledge of the VCLs, interception of a cast vote will reveal nothing about the voter's choice. The VWS can be attacked just like any other Internet-facing web server facility e.g OWASP Top Ten [135], so sensitive vote processing has been located externally in the TSL. VW monitoring by the VW developers will not reveal who an avatar voted for. A malicious PB could send votes to an external RW server, but they would not gain any meaningful information without the corresponding VCL. A compromised VWS cannot send VCLs to unauthorised parties, as they are processed within the TSL. Standardised secure channel protocols (e.g. HTTPs) between the TSL/VWS should give a reasonable level of assurance that information is secure. Using a mobile phone in the voting process provides a more trustworthy channel which is inaccessible to attackers in the

VW: RW phone information is not entered into the insecure VW client so privacy is maintained. The phone also authenticates the VWS and gives the user some assurance that they are dealing with a genuine VWS, as a compromised/ fake VWS would not be able to request VCLs. The VWS/TSL connection could be considered a weak link, so choke points are used to prevent unauthorised access (see Section 7.3.3).

- **Unlinkability (VW2-SR2):** Only a hashed avatar identifier is stored by R in the TSL: the e-VS does not store any avatar details at all. This provides unlinkability between vote contents and avatars. VCLs are not accessible in the VW, so all that can be deduced is that an avatar has voted, but not who they voted for. A human in physical proximity to the user could shoulder surf and link a VCL to a voter. However, the nature of VWs encourages participation on a global scale: it is unlikely (but not impossible) that an attacker is physically present with a RW user. For example, in the 2010 CSM election [299], votes were received from over 20 countries. Thus the threat here should be deemed fairly small. The Registrar, however, can link VCLs to RW phones, a weakness of code voting schemes generally. For higher security elections, this could be addressed by including a secure key generating function in the TSL, and using it to create an ElGamal [308] key pair for each user, to be securely stored on the user's phone at registration (e.g. on the mobile phone SIM). VCLs encrypted using an e-VS ElGamal key can then be encrypted using the user's public key via an ElGamal distributed blinding protocol (described in [247]) without losing any security. The contents of the VCL are not revealed in this protocol.

- **Vote Buying/ Coercion (VW2-SR3):** Remote e-Voting is vulnerable to coercion/vote buying. Here, griefing by malicious avatars can be reported to the VW developers, and as mentioned previously the likelihood of a RW coercion attack is small.

Integrity (SR4/SR5):

- **Recorded as Cast (VW2-SR4):** The use of code voting means that manipulation of a vote's contents will not result in a valid vote. If a vote had been tampered with, either by a compromised VWS/ VWC or an external attacker, this would be detected because the vote confirmation code would not be received correctly. Data in transit is protected by HTTPs for all communication between TSL/VWS and TSL/phone. VCLs are generated in the TSL and sent via the mobile network (which are both trusted) so a compromised VWS cannot modify

them. The nature of code voting means that it will be very difficult for malware to generate valid vote codes [91]: invalid vote codes will be detected through incorrect confirmation codes. If phone malware could display a valid vote code VC_M against ballot choice BC_{M+1} this will be difficult to detect, but this is a sophisticated attack.

- **Counted as Recorded (VW2-SR5):** In code voting the WBB is designed to ensure votes are processed as cast.

Authentication (VW2-SR6/SR7):

- **Only eligible voters can vote (VW2-SR6):** Avatar eligibility is checked on the VWS group database. A compromised activation code at registration may result in an incorrect avatar being associated with Ph_U, but voting integrity is maintained as a rogue avatar cannot vote without VCLs from the phone. (The user can deactivate registered avatars if necessary.) Fake UIDs created by a malicious VWS will be detected by R when requesting VCLs as H(UID) will not match stored values.

- **Voters can vote only once (VW2-SR7):** At the PB, a ballot form is only displayed after the VWS checks if a VCL has been sent and the avatar has not voted before (using *status*). If a malicious PB replays existing/generates extra votes these will also be detected when the VWS checks *status*. A malicious user could create multiple accounts to gain extra votes (the Sybil attack [309]). Incorporating a recurring cost for each additional identity is one of the strategies to address the Sybil attack. Here, a VW-specific maximum number of avatars can be registered with a RW phone: adding more identities requires extra RW equipment and may provide a sufficient financial disincentive. Replaying messages between the VWS and e-VS in the TSL are detected by the e-VS as ID_{VCL} is unique, hard-to-guess and cannot be re-used.

Availability - VW2-SR8

The client-server VW architecture is designed to give high availability and resilience against general DDoS attacks. Stealing the phone gives an attacker VCL details, but to vote they also need VW credentials (not stored on the phone). If a vote is suppressed, either between the VWS/TSL or by a malicious PB, no confirmation code will be received. The user can report such anomalies to the VW developers, as well as in-world griefing activities which prevent an avatar using a PB.

7.3.6 Use Case VW-2: Summary

Security issues in VWs have hindered the adoption of in-world voting, so voting is mostly done via forums/wikis. A VW voting solution has been described that uses code voting processes located in a TSL, external to the VWS, along with VCLs sent to the user's mobile phone. A preliminary security analysis demonstrated that the proposal can deal with many of the specific challenges presented by voting in a VW environment.

Table 7.8: VW Use Cases vs Security Requirements

Security Requirement	VW-1	VW-2
Confidentiality	✓	✓
Integrity	✓	✓
Availability	✓	✓
Authentication	✓	✓
Non-Repudiation	✓	N/A

7.4 Chapter Summary

Two use cases *VW-1: VW Login using the SCWS* and *VW-2: VW Voting* were described in this chapter. *VW-1* used the tamper-resistant security properties of the SCWS installed in a SIM, along with OTPs and geolocation to offer security improvements on the current static username/password authentication that is often used to log-in to VWs. *VW-2* used code voting processes located in an externally-located TSL, along with VCLs sent to the user's RW mobile phone. The security of both use cases was assessed informally against previously defined security requirements.

A summary of how well the security requirements were met for each use case is shown in Table 7.8.

7.5 Related Publications

Two publications have resulted from work on VWs covered in this chapter:

1. L. Kyrillidis, G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Virtual World Authentication using the Smart Card Web Server", in *International Symposium on Security in Computing and Communication' (ISSCC2013)*, pp. 30–41. [9]

2. S. Cobourne, G. Hili, K. Mayes, and K. Markantonakis, "Avatar Voting in Virtual Worlds", in *5th International Conference on Information and Communication Systems (ICICS2014)*, pp. 1–6.[10] (Nominated for Best Paper)

An additional paper about VWs was also published, using material not included in this chapter:

- G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Practical Attacks on Virtual Worlds", in *International Conference on Risks and Security of Internet and Systems (CRiSIS2014)*, pp. 180–195. [11]

Part III

Analysis and Conclusion

Chapter 8

Analysis

Contents

Several of the use cases described in this thesis use the SCWS in their proposed solutions: a more detailed analysis of the security properties of the SCWS and its applications is now given. This is followed by a further discussion of the security of the use cases. The chapter ends by noting implementation issues that relate to the SCWS.

8.1 Security Analysis - SCWS

The main security characteristics of applications installed on an SCWS can be summarised as follows:

- **No Centralised Server:** Distributing applications to individual SIM cards means there is no centralised server to be attacked, hence a single point of failure has been removed. DDoS attacks cannot easily take place because the application is dispersed among many users. So an attacker would have to find out which phones are registered for SCWS applications and then infiltrate a large number of them, in order to have a significant impact on security.

- **Trusted Administration:** The RAS could be considered a single point of failure, but access to it is controlled, and it may have restricted functionality. The RAS runs on MNO/TTP premises, so should be isolated from unauthorised physical access. Insider attacks at the MNO/TTP may be possible, but as the OMA specifications define the RAS as a trusted entity, within the scope of this thesis it is assumed that both the infrastructure and management procedures will be securely provided.

- **Secure Token:** The SCWS, the application and the credentials will be stored in a secure tamper resistant token, the SIM card. SIM defences mitigate against physical and side channel attacks, and in the unlikely event that an attacker circumvents these, they will only gain access to one set of credentials. Attacks therefore are not scalable, as an attacker would have to be in possession of a large number of phones and their SCWS/SIMs to be effective.

- **Standardised Communications:** When using FAP, communication takes place over standard secure communication channels, protected by the HTTPs protocol. HTTPs provides reasonable protection against eavesdropping and man-in-the-middle attacks, and it is the de facto protocol used whenever security is needed on the web. Attacks against HTTPs/TLS are possible e.g. [310, 311, 312, 112], but using HTTPs locally reduces the likelihood of them succeeding.

- **Independent Security Evaluation:** The SIM/SCWS could be subjected to an independent security evaluation, using the internationally recognised Common Criteria framework. This provides security evaluations of IT products [313]. There is a Common Criteria Protection Profile (PP) which applies to the (U)SIM and SCWS (SFR SA, 2011) [314]. A successful evaluation against this PP means that a SIM can withstand potential attacks like DoS on the SCWS from the

phone handset, and unauthorised modifications to SCWS code/ servlets have been identified and mitigated.

8.1.1 Attack Surface of the SCWS

The attack surface of a traditional web server is large. A server often hosts a number of web sites, each of which may be accessed by a large number of users (who may or may not be malicious), and each of these web sites could have multiple input fields (attack points for injection etc), along with back-end databases that store data that need to be protected. Access is either local (the administrator) or remote (users accessing the web sites). Attack points can include vulnerable setup of scripting languages (executing on the web server), inadequate setup of the web server itself, even tools and/or code that reside on the server. A malicious person with physical access to the server may easily be able to access and retrieve any non-encrypted data.

The attack surface of the SCWS is very much reduced in comparison to traditional web servers, due to the following characteristics:

- **Physical protection:** The SCWS web server is physically protected as it executes inside the tamper resistant environment of the SIM card.
- **Trusted Access:** as seen previously, the SCWS is not accessible to any untrusted remote entities. Only applications on the phone that the ACP enforcer authorises and the trusted RAS may access the SCWS, using tightly managed secure protocols and channels, whereas traditional web servers are designed to be accessible by many remote users.
- **Reduced Functionality:** To operate in the resource-limited environment of the SIM card, the SCWS has reduced functionality and storage capacity. The code running will be much smaller, which in itself does not necessarily reduce the attack surface e.g. if the code has been constructed and implemented badly, but if properly implemented smaller code should mean there are fewer ways to attack SCWS applications. Because of the ACP enforcer, it is difficult for malware on the phone to attack the SCWS, but if the ACP enforcer is compromised, reduced functionality will again limit the options for attacks.
- **User Authentication:** Setting a user PIN/Password to allow authentication of individuals accessing the SCWS environment adds further protection. If this password/PIN is retrieved by an attacker, it is not possible to access the SCWS without possession of the phone too.

- **Attack Scalability:** Remote attacks on the SCWS are difficult, due to its trusted access protocols, and remote DDoS attacks are hard to mount. However, a DoS attack against a single SCWS is feasible, as attacks on the SCWS from the phone browser are theoretically possible, but these are not scalable because the attacker would also need physical possession of the handset.

8.1.2 OWASP Top Ten Risks

The Open Web Application Security Project (OWASP) [135] issues a document which includes the Top Ten vulnerabilities that affect the security of web applications [1] .

As the SCWS is a web server, these vulnerabilities could affect the security of SCWS solutions [316]. However, some of the OWASP Top Ten do not apply to SCWS applications.

Table 8.1 shows which of the OWASP Top Ten 2013 apply to the SCWS environment. Vulnerabilities A5 (Security Misconfiguration), A6 (Sensitive Data Exposure) and A9 (Using Known Vulnerable Components) are shown as not relevant to SCWS solutions. This is due to the tamper resistant characteristics of the SIM card, along with the small footprint, minimal functionality and tightly controlled management of the SCWS which should provide assurance that credentials are kept securely, and system components remain up-to-date and configured properly.

The injection and cross site scripting exploits - A1 (Injection), A3 (Cross-Site Scripting (XSS)) and A8 (Cross-Site Request Forgery (CSRF)) - can be limited by strict filtering of input fields in the phone browser. The reduced complexity of the stripped down SCWS also affords some protection against these vulnerabilities. It should be noted, however, that many of the OWASP Top Ten relate more to application design and implementation rather than web server functionality (i.e. A1, A3, A8 and A10). Table 8.2 summarises the SCWS defences against applicable OWASP Top Ten 2013 vulnerabilities and the residual risk from using the SCWS environment.

8.2 Security Analysis - Use Cases

The security of the use cases presented in earlier chapters of this thesis is now revisited. For each use case, security requirements were identified and an informal security analysis was done: when appropriate, formal analysis using Scyther was also performed,

[1] When the research in this thesis was done, the most recent OWASP Top Ten was from 2013: the OWASP Top Ten - 2017 [315] was released in December 2017. In the 2017 version, A4 (Insecure Direct Object References) and A7 (Missing Function Level Access Control) were merged into a new category A5 (Broken Access Control), whilst A8 (CSRF) and A10 (Unvalidated Redirects and Forwards) were retired.

Table 8.1: OWASP Top Ten 2013 and Relevance to the SCWS

Reference	Description	SCWS
A1 Injection	Untrusted data is sent as part of a command or query e.g. SQL (Structured Query Language) or LDAP (Lightweight Directory Access Protocol) that can trick an interpreter into executing unintended commands or access unauthorised data.	Yes
A2 Broken Authentication/ Session Management	Attackers can compromise passwords, keys, or session tokens, or exploit other implementation flaws to assume other users' identities.	Yes
A3 Cross-Site Scripting (XSS)	If an application takes untrusted data and sends it to a web browser without proper validation, attackers can execute scripts in the browser e.g. to hijack sessions, or to redirect unsuspecting users to malicious sites.	Yes
A4 Insecure Direct Object References	A direct object reference concerns an internal implementation object, e.g. file, or directory: poor access control can lead to manipulation of these references by attackers and unauthorised access to data.	Yes
A5 Security Misconfiguration	Applications, frameworks, and servers should have a secure configuration defined and deployed: these must be implemented and maintained (defaults are often insecure), as well as keeping software up to date.	No
A6 Sensitive Data Exposure	Weakly protected data could be tampered with, so sensitive data at rest/ in transit should be encrypted: care should be taken when exchanging data with the browser.	No
A7 Missing Function Level Access Control	Applications must perform verify function level access control checks on the server, otherwise attackers will be able to forge unauthorised functionality requests.	Yes
A8 Cross-Site Request Forgery (CSRF)	A forged HTTP request is sent from a logged-on victims browser to a vulnerable web application that will then be treated as a legitimate request from the victim	Yes
A9 Using Known Vulnerable Components	Applications using components such as libraries, frameworks, and other software modules with known vulnerabilities may result in a number of attacks.	No
A10 Unvalidated Redirects and Forwards	Without proper validation, attackers can redirect users to malicious sites, or use forwards to access unauthorised pages.	Yes

Table 8.2: OWASP Top Ten 2013 - SCWS Defences and Residual Risks

Ref	Defences	Residual Risks
A1	The SCWS does not support SQL or LDAP so SQL/LDAP injection attacks do not apply here.	Non - SQL/LDAP injection attacks targeting input fields may occur. Filtering of input will be needed.
A2	All credentials are stored inside the tamper resistant SIM card, and secure HTTPs connections protect credentials whilst in transit.	Even if data from one phone leaks, this affects one device only, and an attacker would need to be in possession of the phone.
A3	Strict filtering of input fields will be needed.	Limited complexity provides a certain level of assurance.
A4	Applications on the SCWS may have different levels of access (e.g. admin/ user), so proper authorisation and verification of all access requests should be done.	There is nothing in the OMA specification that forbids an SCWS application to have multiple users on one SCWS, but in practice this vulnerability may not apply.
A7	SCWS Applications should not show links to unauthorised functions, and authentication/authorisation checks must be included.	The phone browser is allowed to access a webpage in the SIM card, so attacks may arise (e.g. through phone malware).
A8	The SCWS should be able to implement CSRF countermeasure such as forcing a user to re-authenticate.	The SCWS has restricted functionality which provides added defences.
A10	Redirects should be validated.	Manipulation of the server side (the tamper resistance of the SCWS should protect against this) and the phone browser would be needed, which is a difficult attack to perform successfully.

and no attacks were found within bounds. The results are summarised in Table 8.3. It can be seen that the security requirements were, on the whole, well met.

Solutions presented for the use cases fall into two categories: SCWS-based (i.e. *EV-1/2, MP-1*, *Auth-1* and *VW-1*) and non-SCWS-based (i.e. *MP-2*, *Auth-2* and *VW-2*). These categories are now discussed.

8.2.1 SCWS-based Solutions: EV-1, EV-2, MP-1, Auth-1 and VW-1

Each of these solutions exhibits the following characteristics that enhance security in challenging environments:

- **Identification:** All the presented SCWS solutions provide two-factor authenti-

cation of users, by using a correct PIN/Password for accessing the SCWS environment ("something you know"), in conjunction with possession of the SCWS ("something you have"). Local authentication is possible, which is particularly relevant in use case *Auth-1*. Additionally, use case *EV-2* demonstrated that it is feasible to use the SCWS with National PKI schemes, using Estonian I-voting as an example. Estonia has been called the "most advanced digital society in the world" [317], but there are national PKI schemes being proposed for other (possibly less technologically developed) nations such as Kenya [318] and the Philippines [319], that could consider the SCWS approach for remote e-voting using mobile phones.

- **Connectivity:** To install and update code and data on the SCWS, there must be connectivity: however, much of the other processing can be done offline without loss of security. So, votes can be cast offline in *EV-1/2* and retrieved at a later time, authentication can be done wholly offline in *Auth-1*, and transfer requests can be made and processed at a later time in *MP-1*. However, face-to-face transactions such as withdrawals and deposits need to be communicated to and from the bank in a reasonable time. Offline processing is not relevant to *VW-1*.

- **Attack Resistance:** The small attack surface of the SCWS, the tamper resistance of the SIM, and the resilience against DDoS attacks provided through distributing applications to a large number of SCWS/SIMs all reduce the likelihood of attacks succeeding. Attack resistance is a welcome security feature, especially when technical expertise may be scarce in the relevant operating environment. This applies to all the use case solutions in this category.

8.2.2 Non-SCWS-based Solutions: MP-2, Auth-2 and VW-2

The three remaining use cases did not use the SCWS in their solutions.

The Bitcoin m-payment solution *MP-2* was designed to bring the security of distributed ledger transactions to areas with the minimum available connectivity i.e. SMS messaging only. It would be possible to use the SCWS to store credentials, and process transactions on the SIM as in use case *MP-1*, by issuing advanced SCWS/SIMs instead of the OTP token. The SCWS could generate OTPs as required, and send transaction details via secure HTTPs communication and the RAS. However, SCWS transactions need some online access to communicate between the RAS and the SCWS, so this may not be a suitable option in the targeted environment. Also, the need to have a business relationship with the MNO that owns the SIM would reduce the immediate deployability/ interoperability of the m-payment scheme in humanitarian aid scenarios,

where time is of the essence[2]. The presented solution using SMS provides a pragmatic balance between security and usability.

In use case *VW-2* there is minimal input from the mobile phone - it is used as a mere conduit for voting credentials, via a mobile application. It would be possible to employ an SCWS solution for storing and processing votes/voting credentials as in *EV-1/EV-2*. However, this would move the voting "ceremony" to the RW as the votes would then be input to the RW phone, rather than as an in-world voting activity. Sending VCLs via the MNO network provides a second channel inaccessible from the VW, so protects against in-world attacks: placing sensitive vote processing in a TSL with a small attack surface, that is accessed using secure communication protocols (e.g. HTTPs) introduces a trustworthy component to the proposal. Even if the VCLs are obtained from the potentially insecure mobile phone platform, knowledge fragmentation across four distinct zones means that they cannot be used to mount an attack without information from the other three zones, thus increasing the complexity of a potential attack. Again, a balance between security and usability is struck.

As *Auth-2* was concerned with using depth cameras to create a dynamic biometric based on gesture recognition, the SCWS is not relevant. It may be possible in future to interface the authentication results from gesture recognition with the SCWS but this would need further research. Gesture recognition has the potential to become a flexible, non-intrusive, non-contact method for local authentication, with the advantage that a gesture is changeable in the case of compromise. In challenging environments this would provide a useful alternative authentication method, especially as dynamic biometry also improves unlinkability. The feasibility of match-on-card DTW processing of gestures captured using the Kinect/ Leap Motion devices was assessed: this could be performed on a phone using an HCE application [8]. In future, as 3D depth cameras become more widely available on mobile devices, the need for separate equipment to capture data will be removed, so this dynamic biometric authentication method could be used directly with phone sensors. However, research in this area is still at a very early stage.

8.3 SCWS Implementation Issues

Proof of concept versions for both the generic e-voting solution and the branchless banking application were produced in a simulated environment and successfully tested on a laptop computer (hosting an AMD processor 2.4GHz with 2GB of RAM and 512

[2]Instead of OTP tokens, providing SIMs with a SIM Toolkit m-payment application installed could be an alternative: again this would require a business relationship with the MNO, which may reduce the potential speed of deployment.

Table 8.3: Security Requirements for Each Use Case

Security Req.	EV-1	EV-2	MP-1	MP-2	Auth-1	Auth-2	VW-1	VW-2
Confidentiality	Part[a]	✓[b]	✓	Part[d]	✓	N/A	✓	✓
Integrity	✓	×[c]	✓	Part[d]	✓	N/A	✓	✓
Availability	✓	✓	✓	✓	✓	N/A	✓	✓
Authentication	✓	✓	✓	Part[d]	✓	Part[e]	✓	✓
Non-Repudiation	N/A	N/A	✓	✓	✓	N/A	✓	N/A

[a] Not Coercion Resistant e-voting scheme
[b] Re-voting allowed as anti-coercion measure.
[c] Not Voter Verifiable e-voting scheme (pre-2015)
[d] SMS messages are not confidential/can be spoofed
[e] The Kinect experiment fully meets these requirements: the Leap Motion is less accurate

GB of hard drive): the code was created as a JavaCard 3.0 Connected Edition project running on a simulated platform provided by NetBeans IDE. However, somewhat disappointingly, the SCWS has not generally been implemented in real SIMs. Despite the best efforts of the SIMAlliance and OMA in promoting SCWS functionality, the fact that the SCWS is designed to be installed on the SIM raises barriers to adoption, as the MNO owns the platform. This raises ecosystem issues, similar to those that were encountered when NFC technology was first proposed for m-payment applications.

The card-emulation mode available on NFC phones could be used for contactless payments using financial credentials stored on an SE, but there was much debate about the best place to locate this SE. The options are a) on an embedded SE on the phone (i.e. owned by the handset manufacturer), b) on the SIM (i.e. a SIM-SE owned by the MNO) or c) on a removable SE (owned by the user) [320]. The resulting competing management issues caused a barrier to adoption of NFC for use with mobile payments. Google introduced an HCE solution (in software rather than hardware) for Android Pay [321]: HCE allows developers a quicker route to market for their schemes, albeit without the tamper resistant security of a hardware chip. However, Apple decided to use the embedded SE route to store financial credentials and tokens for their Apple Pay m-payment scheme [113], and NFC m-payment schemes are now more widespread. By liaising with handset manufacturers, financial institutions can more easily "buy-in" to the scheme and do not need their own management procedures for direct access to the SE chip. This is a hurdle the SCWS has yet to overcome.

Another implementation issue that affects the SCWS is that it needs support from handset manufacturers to provide modifications to the mobile operating system to incorporate the required BIP gateway. An open source repository, the SEEK for Android project [322], previously included SCWS features, but they are no longer available at

time of writing. The SCWS might also benefit from the proposal for a high powered USB contact on the chip, but the lack of available real examples hinders testing. The other communication option, via HTTPs and a TCP/IP stack, will not be possible until SIM cards able to support JavaCard 3.0 Connected Edition become available.

This situation could change in future, as was seen with the NFC m-payment ecosystem: a "killer app" or a supportive manufacturer could alter the landscape. Alternatively, the functionality and desirable security properties of the SCWS may become available in other security devices/chips, in which case the solutions presented in this thesis will contribute secure applications for this new ecosystem.

8.4 Chapter Summary

This chapter provided a more in-depth analysis of the security features of the SCWS and its applications. In particular, the SCWS defences against relevant OWASP Top Ten 2013 vulnerabilities, and the residual risk of using the SCWS environment were identified. The security analyses of the use cases that had been presented in earlier chapters were then extended. A discussion followed which noted implementation issues for SCWS solutions.

Chapter 9

Conclusion and Future Work

Contents

This chapter summarises the work presented in this thesis and the contributions made, and assesses how well the research aims have been met. Future research directions are then identified and the thesis is concluded.

9.1 Summary and Conclusion

The main goal of this work was to to design solutions using mobile devices that would enhance security in challenging environments, and secondly to design improved authentication methods that can be used in both RW and VW scenarios. Security solutions from three RW application areas were proposed, covering a good spread of challenging environments, and then two of these areas were investigated in the VW. A summary of the application areas and use cases now follows.

9.1.1 Application Area: e-Voting

Chapter 4 introduced the **Remote e-Voting** application area, which has been the subject of extensive academic research. However, schemes which use mobile phones for voting are not easy to find, due to the difficulty in ensuring that the mobile device will process votes in the correct manner (the secure platform problem). Using a phone for voting would have great benefit in areas where it may be dangerous or practically impossible to reach an electoral poll site, for example in remote communities, or due to physical immobility of voters. A "front-end" SCWS generic e-voting model was proposed, and two e-voting schemes, Prêt à Voter and Estonian I-voting, were used as examples of its applicability in use cases *EV-1* and *EV-2* respectively. The solution presented distributed web server functionality to voters' SIMs, so that there was no central vote-processing web server to target: an attacker would need to compromise many phones to successfully affect the election result. The use of the SIM's tamper-resistant environment for the storage and processing of sensitive voter credentials also addresses the secure platform problem. Thus the principle of using a ubiquitous device (the phone) with an SCWS application that provides a secure distributed architecture for remote e-voting was established. The SCWS voting application protects the e-voting system by making the effort required to attack vote casting prohibitively high.

CONTRIBUTIONS:
EV-1/EV-2: The SCWS was used in a solution that provides tamper resistance and protection against DDoS attacks in remote e-voting, which was illustrated using e-voting systems Prêt à Voter and Estonian I-voting as examples [1] [2].

9.1.2 Application Area: m-Payment

Chapter 5 covered the **M-Payment** application area, where there are many existing solutions (and corresponding security issues) as discussed in [27].

In use case *MP-1*, a branchless banking scheme was presented that used PKI-capable SIMs equipped with a SCWS to process withdrawals, deposits and transfers in a secure and user-friendly manner. Even though these specialised SIMs are more expensive than conventional SIMs, this is a cheaper overall solution than setting up physical bank branches. A preliminary security analysis indicated that the security of this proposal is higher than that offered by the most widely used m-payment scheme in the developing world, M-PESA.

All systems present a trade-off between usability and security, and the second use case in this area, *MP-2*, presented a pragmatic solution where infrastructure constraints limit the security options available. It described how Bitcoin transactions could be made in an area where Internet connectivity is not available, that would enable a charitable organisation to provide humanitarian aid in Bitcoin. The proposal included hosted Bitcoin wallets maintained by the charity, an SMS based mobile payment system and an OTP token-based two-factor authentication method.

CONTRIBUTIONS:
MP-1: The SCWS and its tightly managed, standardised management protocols were used in a branchless banking application, to provide enhanced security compared to other SIM-based m-Payment schemes, such as M-PESA [3].
MP-2: Access to secure blockchain technology was enabled via an SMS m-payment system, for use by charitable organisations in offline humanitarian aid scenarios [4].

9.1.3 Application Area: Authentication

Chapter 6 discussed **Authentication** techniques. Use case *Auth-1* proposed a Single Sign-On (SSO) solution for disconnected environments by using the tamper-resistant security properties of the SCWS installed in a SIM (SIM-SCWS) with in another SCWS smart card chip (MOD-SCWS) embedded within an electronic assembly. Local authentication on the mobile device SIM-SCWS produces a security token that is sent to the MOD-SCWS over local wireless channels communications. The distributed authentication approach avoids a single point of failure i.e. a centralised SSO server, and as seen before, attacks against an individual SCWS are not scalable as they require physical possession of the SIM-SCWS or MOD-SCWS.

In contrast, use case *Auth-2* investigated the potential for using depth cameras on a smart phone as sensors for dynamic biometric authentication, using the DTW algorithm to analyse the captured data. As depth camera phones are not currently commercially available, preliminary experiments were done using Kinect and Leap Motion devices to assess the accuracy and practicality of the approach, with promising

results.

CONTRIBUTIONS:
Auth-1: Secure offline authentication in an SSO application was facilitated by using the SCWS (installed in both a SIM and a security module) and near field communications to exchange security tokens [6].
Auth-2: An investigation into the feasibility of using gesture recognition as a two-factor one-step dynamic biometric authentication method was carried out [5, 7, 8].

9.1.4 Application Area: VW Applications

Chapter 7 took two of the previously studied application areas (authentication and e-voting) and applied them to the VW environment as **VW Applications**. *VW-1* again used the SCWS for authentication, but this time in conjunction with OTPs and geolocation to offer security improvements on the current static username/password authentication that is often used for VW login. Use case *VW-2* provided secure and private in-world voting by locating code voting processes in a trusted external zone, the TSL: VCLs were sent to the user's RW mobile phone in order to complete the voting process.

CONTRIBUTIONS:
VW-1: Authentication of online log-in to VWs was enhanced using the SCWS, geolocation and OTP processes [9].
VW-2: Privacy was introduced into a VW e-voting application via knowledge fragmentation across four distinct zones, using a mobile phone to receive security information over a second channel i.e. the MNO network [10].

9.1.5 SCWS Solutions

The SCWS was used in five of the solutions presented: the main strength of using the SCWS is that it uses standardised hardware, protocols and communications to protect sensitive information, without the need for specialised equipment and phone applications. By storing security information on the tamper-resistant SIM, local authentication can be done by the SCWS without communicating credentials across a network. Using existing tamper-resistant hardware (the SIM) with standardised features (SCWS) along with the MNO's FAP (via HTTPs), means that sensitive information can be protected at all times. The security of these standardised elements has been exten-

sively investigated by the expert community. It is difficult to mount large scale attacks against the proposed solutions, as credentials and applications stored on each SIM must be targeted individually. The SCWS has not been implemented widely yet, but the solutions presented could contribute secure applications if SCWS functionality was to become available in other security devices/chips in future.

9.1.6 Meeting Research Objectives

The chosen methodology of proposing solutions for use cases in three application areas has led to the investigation of a wide range of security challenges. Having analysed the relative successes of each solution with respect to stated security requirements for each use case in Section 8.2, it is possible to assess how well the research objectives/ research questions outlined in Chapter 1 have been met i.e.

1. Research Questions

    - **RQ-1:** How can introducing a trustworthy infrastructure based on mobile devices and/or alternative methods of authentication address the differing security requirements of a range of use cases in these environments?
    - **RQ-2:** How can a trusted element in a mobile device be used to enhance security in challenging environments, where there may be limited access to technical infrastructure and resources?

2. Research Objectives

    - **RO-1:** (Main Objective) Design security solutions using mobile devices to enhance security in a range of use cases in challenging environments.
    - **RO-2:** (Secondary Objective) Design improved authentication methods that can be used in challenging environments.

Table 9.1 shows which research questions and objectives have been addressed by which use cases. (Use case *Auth-2* is shown as "Part" because the work described is still at a preliminary stage.) It can be seen that by using a good spread of use cases the aims of the thesis have been met.

9.2 Future Work

There are many areas which could benefit from further research, some of which are outlined now.

Table 9.1: Research Questions/Objectives vs Use Cases

RQ/RO	EV-1	EV-2	MP-1	MP-2	Auth-1	Auth-2	VW-1	VW-2
RQ-1	✓	✓	✓	✓	✓	✓	✓	✓
RQ-2	✓	✓	✓		✓		✓	
RO-1	✓	✓	✓	✓	✓	Part	✓	✓
RO-2	✓	✓	✓		✓	✓	✓	

As seen in Section 8.3 the Smart Card Web Server has not been widely adopted in the SIM environment at the time of writing, so one of the major research directions could be to investigate if its functionality and desirable security properties could be applied to other security devices/chips in future. In particular, based on the work presented, it would be interesting to investigate the possibility of creating a research platform that abstracts away from the Java card, which could be used with alternative Trusted Execution Environments. If this became a reality, practical implementations of the solutions presented would enable performance measurements to be taken, to give an indication how speed of processing will impact their usability/ security, and suitability for use with various phone handsets.

In this case (or if the SCWS becomes more widely available generally), the SCWS solutions proposed in this thesis could be extended in a number of ways. For example, the generic e-Voting model described in Chapter 4 could be amended to process votes from multiple voters. Here, each voter would need personalised ballot forms on the SCWS, accessible to them only once they are suitably authenticated. Confirmation SMS messages from the voting authority could go to a nominated phone number, not necessarily back to the SCWS used to vote. Since the SCWS does not need to be online to process votes, the proposed SCWS voting method could be used as a portable voting booth in situations with no network connectivity: votes could then be uploaded to the e-voting system at a later stage, perhaps via a transfer point in a controlled voting kiosk area, or once connectivity is restored. Also, investigating measures that could minimise the involvement of the MNO in the voting process would be worthwhile.

The SCWS branchless banking approach from use case *MP-1* could be extended to interface with the Bitcoin processes detailed in use case *MP-2*, for use in areas with more than minimal connectivity: this would enhance the security of the Bitcoin m-payment proposal. Additionally, devising NFC m-payments schemes that use the SCWS could provide an interesting research direction.

Application of "thin SIM" technology and its security could merit further investigation: m-payment solutions have been suggested that use these stick-on Overlay SIMs (e.g. [323, 324]) but the GSMA have cautioned that there are security issues - such as

trojans, eavesdropping and unauthorised access to SIM card configuration settings - that first need to be addressed [325].

Investigating how to interface the SCWS with authentication results from gesture recognition performed on a phone could merit further research, along with testing implementations of SE/HCE match-on-card applications that use other feature extraction algorithms as well as DTW. When 3D depth cameras become available on phones, future work on gesture recognition biometric authentication on phones could be undertaken, as current research in this area is still at an early stage.

As testing applications in RW challenging environments can be difficult and dangerous, a study into the feasibility of creating a VW tool that could be trained and optimised to detect insecure actions in VW models of RW scenarios could be useful. VWs are providing new features that will make the immersive experience more realistic, such as the inclusion of Virtual Reality functionality in SL's forthcoming new social world, "Sansar" [326], that could be used with gesture recognition authentication and add value to VW security modelling.

Bibliography

[1] L. Kyrillidis, S. Cobourne, K. Mayes, S. Dong, and K. Markantonakis, "Distributed e-voting using the Smart Card Web Server," in *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS)*. IEEE, 2012, pp. 1–8.

[2] S. Cobourne, L. Kyrillidis, K. Mayes, and K. Markantonakis, "Remote e-voting using the Smart Card Web Server," *International Journal of Secure Software Engineering (IJSSE)*, vol. 5, no. 1, pp. 39–60, 2014.

[3] S. Cobourne, K. Mayes, and K. Markantonakis, "Using the Smart Card Web Server in Secure Branchless Banking," in *International Conference on Network and System Security (NSS2013)*. Springer, 2013, pp. 250–263.

[4] D. Jayasinghe, S. Cobourne, K. Markantonakis, R. N. Akram, and K. Mayes, "Philanthropy On The Blockchain," in *11th WISTP International Conference on Information Security Theory and Practice (WISTP2017)* , 2017.

[5] B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Authentication based on a changeable biometric using gesture recognition with the Kinect," in *2015 International Conference on Biometrics (ICB)*. IEEE, 2015, pp. 38–45.

[6] L. Kyrillidis, S. Cobourne, K. Mayes, and K. Markantonakis, "A Smart Card Web Server in the Web of Things," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2016, pp. 769–784.

[7] B. Ducray, S. Cobourne, K. Mayes, and K. Markantonakis, "Comparison of dynamic biometric security characteristics against other biometrics," in *IEEE International Conference on Communications (IEEE ICC 2017)*. IEEE ICC 2017, May 2017.

[8] ——, "Gesture recognition implemented on a personal limited device," in *The International Conference on Information and Communication Systems (ICICS 2017)*, 2017.

[9] L. Kyrillidis, G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Virtual World Authentication Using the Smart Card Web Server," in *Security in Computing and Communications*, ser. Communications in Computer and Information Science, S. Thampi, P. Atrey, C.-I. Fan, and G. Perez, Eds. Springer Heidelberg, 2013, vol. 377, pp. 30–41.

[10] S. Cobourne, G. Hili, K. Mayes, and K. Markantonakis, "Avatar voting in Virtual Worlds," in *Information and Communication Systems (ICICS), 2014 5th International Conference on*. IEEE, 2014, pp. 1–6.

[11] G. Hili, S. Cobourne, K. Mayes, and K. Markantonakis, "Practical attacks on Virtual Worlds," in *International Conference on Risks and Security of Internet and Systems*. Springer, 2014, pp. 180–195.

[12] A. Tomlinson and S. Cobourne, *Smart Cards, Tokens, Security and Applications, 2nd Edition*, 2nd ed. Springer, 2017, ch. 6: Security for Video Broadcasting, pp. 155–171.

[13] J. Cadonau, D. Jayasinghe, and S. Cobourne, *Smart Cards, Tokens, Security and Applications, 2nd Edition*, 2nd ed. Springer, 2017, ch. 11: OTA and Secure SIM Lifecycle Management, pp. 283–304.

[14] K. Mayes, S. Cobourne, and K. Markantonakis, "Near Field Technology in Challenging Environments," *Smart Card Technology International. NFC and Contactless*, pp. p. 65–69, 2011.

[15] Open Mobile Alliance. (2013) OMA Smartcard Web Server V1.2. Accessed: September 2017. [Online]. Available: http://www.oma-works.org/Technical/release_program/scws_v1_2.aspx

[16] Microsoft. (2016) Kinect Sensor. Microsoft. Accessed: September 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/hh438998.aspx

[17] D. Chaum, "Surevote: technical overview," in *Proceedings of the workshop on trustworthy elections (WOTE01)*, 2001.

[18] P. Ryan and V. Teague, "Pretty Good Democracy," in *Workshop on Security Protocols*, vol. 154, 2009.

[19] State Electoral Office of Estonia. (2017, June) General Framework of Electronic Voting and Implementation thereof at National Elections in Estonia. Accessed: September 2017. [Online]. Available: http://www.vvk.ee/public/EHS/IVXV-UK-1.0-eng.pdf

[20] S. Thapa. (2014, July) Five Smartphones for Under $50 USD. [Online]. Available: http://techchange.org/2014/07/30/cheap-smartphones-under-50-dollars/

[21] GSMArena. Phone Finder. Accessed: September 2017. [Online]. Available: http://www.gsmarena.com/search.php3?

[22] P. Norris, *Digital divide: Civic engagement, information poverty, and the Internet worldwide*. Cambridge University Press, 2001.

[23] International Telecommunications Union (ITU). (Visited, October 2016) ICT Facts And Figures 2016. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf

[24] Apple Inc. (2017) Use Touch ID on iPhone and iPad. Accessed: September 2017. [Online]. Available: https://support.apple.com/en-gb/HT201371

[25] Samsung. (2017) How do I use the fingerprint scanner on my Samsung Galaxy Tab S? Accessed: September 2017. [Online]. Available: http://www.samsung.com/us/support/answer/ANS00044932/

[26] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on.* IEEE, 2009, pp. 641–644.

[27] B. Reaves, N. Scaife, A. M. Bates, P. Traynor, and K. R. Butler, "Mo (bile) Money, Mo (bile) Problems: Analysis of Branchless Banking Applications in the Developing World." in *USENIX Security*, 2015, pp. 17–32.

[28] Blizzard Entertainment Inc. (2017) Battle.net Authenticator. Accessed: September 2017. [Online]. Available: https://eu.battle.net/support/en/article/24520

[29] K. E. Mayes and K. Markantonakis, *Smart Cards, Tokens, Security and Applications 2nd Edition*, K. E. Mayes and K. Markantonakis, Eds. Springer, 2017.

[30] C. J. Cremers, "The Scyther tool: Verification, falsification, and analysis of security protocols," in *CAV*, vol. 8. Springer, 2008, pp. 414–418.

[31] J. Casswell. (2017, February) Displaced Populations, Humanitarian Cash Transfers and Mobile Money. GSMA. [Online]. Available: http://www.gsma.com/mobilefordevelopment/programme/mobile-money/displaced-populations-humanitarian-cash-transfers-and-mobile-money

[32] GSMA. (2012) Dealing with disasters: Technical challenges for mobile operators. Accessed: September 2017. [Online]. Available: http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2012/06/Dealing-with-Disasters_Final.pdf

[33] S. Goodman and A. Harris, "The coming African tsunami of information insecurity," *Communications of the ACM*, vol. 53, no. 12, pp. 24–27, 2010.

[34] L. M. Angela Crandall, Albert Otieno, J. G. Jessica Colao, and P. Otieno. (2012, December) Mobile Usage at the Base of the Pyramid in Kenya. International Bank for Reconstruction and Development/ World Bank. Accessed: September 2017. [Online]. Available: https://www.infodev.org/infodev-files/final_kenya_bop_study_web_jan_02_2013_0.pdf

[35] I. Medhi, S. Patnaik, E. Brunskill, S. Gautama, W. Thies, and K. Toyama, "Designing mobile interfaces for novice and low-literacy users," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 18, no. 1, p. 2, 2011.

[36] I. Medhi, S. Gautama, and K. Toyama, "A comparison of mobile money-transfer uis for non-literate and semi-literate users," in *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009, pp. 1741–1750.

[37] AfroBarometer. (2015, September) Devolution and corruption in Kenya. Accessed: September 2017. [Online]. Available: http://afrobarometer.org/blogs/devolution-and-corruption-kenya

[38] GSMA. (2017) Mobile for Development. GSMA. Accessed: September 2017. [Online]. Available: https://www.gsma.com/mobilefordevelopment/

[39] United Nations Foundation. Compendium of m-health projects. United Nations Foundation. Accessed: September 2017. [Online]. Available: http://www.globalproblems-globalsolutions-files.org/unf_website/assets/publications/technology/mhealth/mHealth_compendium_full.pdf

[40] C. Danis, J. Ellis, W. Kellogg, H. van Beijma, B. Hoefman, S. Daniels, and J. Loggers, "Mobile phones for health education in the developing world: SMS

as a user interface," in *Proceedings of the First ACM Symposium on Computing for Development*. ACM, 2010, p. 13.

[41] A. Marcus, G. Davidzon, D. Law, N. Verma, R. Fletcher, A. Khan, and L. Sarmenta, "Using NFC-enabled mobile phones for public health in developing countries," vol. 2009. Institute of Electrical and Electronics Engineers, 2009, pp. 30 – 35, accessed: September 2017. [Online]. Available: http://hdl.handle.net/1721.1/59986

[42] D. Holstius, J. Kaye, and E. Seto, "Wireless monitoring of a distributed environmental health intervention in Haiti," in *Wireless Health 2010*. ACM, 2010, pp. 204–205.

[43] C. Brown. NFC phones help provide clean water to Haiti earthquake victims. Accessed: September 2017. [Online]. Available: http://www.nearfieldcommunicationsworld.com/2011/03/11/36414/nfc-phones-help-provide-clean-water-to-haiti-earthquake-victims/

[44] (2017) Using Cell Phones for data entry to a Dengue fever decision support system. Colorado State University. Accessed: September 2017. [Online]. Available: http://www.cs.colostate.edu/ddss/index.html

[45] A. Zalzala, S. Chia, L. Zalzala, and A. Karimi, "Healthcare technologies in developing countries," in *GCC Conference and Exhibition (GCC), 2011 IEEE*. IEEE, 2011, pp. 629–632.

[46] S. Chia, A. Zalzala, L. Zalzala, and A. Karim, "Intelligent Technologies for Self-Sustaining, RFID-Based, Rural e-Health Systems," *IEEE Technology and Society Magazine*, vol. 32, no. 1, pp. 36–43, 2013.

[47] S. Chia, A. Zalzala, L. Zalzala, and A. Karimi, "RFID and Mobile Communications for Rural e-Health: A Community Healthcare System Infrastructure Using RFID for Individual Identity," in *Global Humanitarian Technology Conference (GHTC), 2011 IEEE*. IEEE, 2011, pp. 371–376.

[48] SafariCom. (2017) SafariCom. Accessed: September 2017. [Online]. Available: http://www.safaricom.co.ke/

[49] S. Panjwani and E. Cutrell, "Usably secure, low-cost authentication for mobile banking," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*. ACM, 2010, p. 4.

[50] B. Arora and A. Metz Cummings, "A Little World: Facilitating safe and efficient m-banking in rural India," GIM Case Study No. B051. New York: United Nations Development Programme, 2010.

[51] A. Sharma, L. Subramanian, and D. Shasha, "Secure branchless banking," in *ACM SOSP Workshop on Networked Systems for Developing Regions (NSDR)*, 2009.

[52] A. Karunanayake, K. De Zoysa, and S. Muftic, "Mobile ATM for developing countries," in *Proceedings of the 3rd international workshop on Mobility in the evolving internet architecture*, ser. MobiArch '08. New York, NY, USA: ACM, 2008, pp. 25–30.

[53] mChek. (2012) mChek. Accessed: 2012 - not available September 2017. [Online]. Available: http://main.mchek.com/

[54] P. White, F. Ellis, S. Devereux, and K. Vincent, "Electronic delivery of social cash transfers," *Frontiers of Social Protection Brief*, 2010. [Online]. Available: http://www.kulima.com/wp-content/uploads/2011/03/FOSP-BRIEF-3-Web.pdf

[55] J. H. Paul Harvey, Katherine Haver and B. Murphy. (2010) Delivering money: Cash transfer mechanisms in emergencies. by Save the Children UK. Cash Learning Partnership. Accessed: September 2017. [Online]. Available: http://www.cashlearning.org/downloads/delivering-money---cash-transfer-mechanisms-in-emergencies2.pdf

[56] ALNAPInnovations. (2009, August) Cash transfers through mobile phones: an innovative emergency response in Kenya. Active Learning Network for Accountability and Performance in Humanitarian Action (ALNAP) . Accessed: September 2017. [Online]. Available: http://www.alnap.org/pool/files/innovationcasestudyno1-concern.pdf

[57] S. Muzammil. (2011, May) Vouchers Keep Food Moving despite Syrian Unrest. World Food Programme. Accessed: September 2017. [Online]. Available: www.wfp.org/stories/vouchers-keep-food-moving-despite-syrian-unrest

[58] A. Azfar, J. Jiang, L. Shan, M. J. P. Marval, R. Yanggratoke, and S. Ahmed, "ByteWalla: Delay Tolerant Networks on Android phones," *CSD Labs, The Royal Institute of Technology, Stockholm, Final Report*, 2010.

[59] H. Ntareme, M. Zennaro, and B. Pehrson, "Delay tolerant network on smartphones: applications for communication challenged areas," in *Proceedings of the*

3rd Extreme Conference on Communication: The Amazon Expedition. ACM, 2011, p. 14.

[60] M. Hognerud, P. Sjödin, B. Pehrson, H. Ntareme, D. Gligoroski, D. Avri, and M. Zennaro, "Bytewalla IV Implementation of Delay Tolerant Networks on the Android platform," *KTH-Royal Institute of Technology*, 2010.

[61] H. Ntareme and S. Domancich, "Security and performance aspects of Bytewalla: A Delay Tolerant Network on smartphones," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*. IEEE, 2011, pp. 449–454.

[62] M. Hognerud, "Bytewalla IV: Routing and Application Layer Optimizations for Delay-Tolerant Networks," Master's thesis, KTH-Royal Institute of Technology, 2011.

[63] K. Scott and S. Burleigh. (2007, November) Rfc 5050 - bundle protocol specification. Experimental Protocol. [Online]. Available: https://tools.ietf.org/html/rfc5050

[64] P. Gardner-Stephen, "The Serval project: Practical wireless ad-hoc mobile telecommunications," *Flinders University, Adelaide, South Australia, Tech. Rep*, 2011.

[65] P. Gardner-Stephen, R. Challans, J. Lakeman, A. Bettison, D. Gardner-Stephen, and M. Lloyd, "The Serval mesh: A platform for resilient communications in disaster & crisis," in *Global Humanitarian Technology Conference (GHTC), 2013 IEEE*. IEEE, 2013, pp. 162–166.

[66] R. J. Anderson and S. J. Bezuidenhout, "Cryptographic credit control in prepayment metering systems," in *Security and Privacy, 1995. Proceedings., 1995 IEEE Symposium on*. IEEE, 1995, pp. 15–23.

[67] I. Medhi, A. Ratan, and K. Toyama, "Mobile-banking adoption and usage by low-literate, low-income users in the developing world," in *International Conference on Internationalization, Design and Global Development*. Springer, 2009, pp. 485–494.

[68] GSMA. (2017, September) Enabling access to mobile services for the forcibly displaced. Accessed: September 2017. [Online]. Available: https://www.gsma.com/mobilefordevelopment/programme/digital-identity/enabling-access-mobile-services-forcibly-displaced

[69] Biometrics: Friend or foe of privacy? Privacy International. [Online]. Available: https://www.privacyinternational.org/sites/default/files/Biometrics_Friend_or_foe.pdf

[70] BBC. (2014, June) Mozilla to sell $25 Firefox OS smartphones in India. Accessed: September 2017. [Online]. Available: http://www.bbc.co.uk/news/technology-27793464

[71] S. Satpathy. (2014, June) This is India's cheapest Android smartphone. Accessed: September 2017. [Online]. Available: http://www.bgr.in/news/this-is-indias-cheapest-android-smartphone/

[72] Staff Writer. (2013, December) MTN Steppa most affordable high performance smartphone ever. Accessed: September 2017. [Online]. Available: https://mybroadband.co.za/news/smartphones/93383-mtn-steppa-most-affordable-high-performance-smartphone-ever.html

[73] Android One. (2017) Google Inc. Accessed: September 2017. [Online]. Available: http://www.android.com/one/

[74] Android Go. (2018) Introducing Android Oreo (Go edition). Accessed March 2018. [Online]. Available: https://www.android.com/versions/oreo-8-0/go-edition/

[75] GSMA Mobile World Congress February 2018. Accessed March 2018. [Online]. Available: https://www.mobileworldcongress.com/about/

[76] J. Kastrenakes. (2018, March) Android Go is here to fix super cheap phones. Accessed March 2018. [Online]. Available: https://www.theverge.com/circuitbreaker/2018/3/1/17052912/what-is-android-go

[77] A. Boxall. (2018, February) You need to know how Android Oreo Go Edition is different from Android One. Accessed March 2018. [Online]. Available: https://www.digitaltrends.com/mobile/what-is-android-oreo-go-edition/

[78] MDN Web Docs: Mozilla. (2015) Firefox OS. Mozilla Developer Network. Archive accessed: September 2017. [Online]. Available: https://developer.mozilla.org/en-US/docs/Archive/B2G_OS

[79] MDN Web Docs. (2015) Mozilla Flame Phone. Accessed: September 2017. [Online]. Available: https://developer.mozilla.org/en-US/docs/Archive/B2G_OS/Phone_guide/Flame

[80] M. Karlsson, G. Penteriani, H. Croxson, A. Stanek, R. Miller, D. Pema, and F. Chitiyo. (2017, July) Accelerating affordable smartphone ownership in emerging markets. Accessed March 2018. [Online]. Available: https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2017/07/accelerating-affordable-smartphone-ownership-emerging-markets-2017.pdf

[81] P. Ryan, D. Bismark, J. Heather, S. Schneider, and Z. Xia, "Prêt àVoter: a voter-verifiable voting system," *Information Forensics and Security, IEEE Transactions on*, vol. 4, no. 4, pp. 662–673, 2009.

[82] D. Sandler, K. Derr, and D. Wallach, "VoteBox: a tamper-evident, verifiable electronic voting system," in *Proceedings of the 17th conference on Security symposium*. USENIX Association, 2008, pp. 349–364.

[83] A. Appel, M. Ginsburg, H. Hursti, B. Kernighan, C. Richards, G. Tan, and P. Venetis, "The New Jersey voting-machine lawsuit and the AVC Advantage DRE voting machine," in *Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections*. USENIX Association, 2009, pp. 5–5.

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

[85] Z. Xia, S. Schneider, J. Heather, P. Ryan, D. Lundin, and P. Howard, "Prêt à Voter: All-in-one," 2007.

[86] B. Randell and P. Ryan, "Voting technologies and trust," *Security & Privacy, IEEE*, vol. 4, no. 5, pp. 50–56, 2006.

[87] Valimised. (2017) Internet Voting in Estonia. Accessed: September 2017. [Online]. Available: http://www.vvk.ee/voting-methods-in-estonia/engindex/

[88] Geneva State Chancellery. Uncovering the veil on Geneva's Internet voting solution. Geneva Information Technology Centre. Accessed 2013. [Online]. Available: http://www.geneve.ch

[89] R. Rivest, "Electronic voting," in *Financial Cryptography*, vol. 1, 2001, pp. 243–268.

[90] R. Oppliger, "How to address the secure platform problem for remote internet voting," *SIS*, vol. 2, pp. 153–173, 2002.

[91] J. Helbach and J. Schwenk, "Secure internet voting with code sheets," in *Proceedings of the 1st international conference on E-voting and identity*. Springer-Verlag, 2007, pp. 166–177.

[92] J. Helbach, J. Schwenk, and S. Schage, "Code voting with linkable group signatures," in *EVOTE08, 3rd International Workshop on Electronic Voting*, 2008.

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

[94] O. Tammik. (2011, July) Electoral Committee Chief Apologizes for Website Collapse. Estonian Public Broadcasting. Accessed: September 2017. [Online]. Available: http://news.err.ee/politics/0e335eef-9dd4-4c8c-a104-f34dd4b778c3

[95] Office for Democratic Institutions and Human Rights. (2011, March) Estonia Parliamentary Elections: OSCE/ODIHR Election Assessment Mission Final Report. Accessed: September 2017. [Online]. Available: http://www.osce.org/odihr/77557

[96] S. Wolchok, E. Wustrow, D. Isabel, and J. Halderman, "Attacking the Washington, D.C. Internet Voting System," *16th Conference of Financial Cryptography and Data Security*, 2012.

[97] L. Payton. (2012, March) NDP voting disruption deliberate, hard to track: More than 10,000 computers used in denial of service attack, voting company Scytl says. CBC/Radio Canada. Accessed: September 2017. [Online]. Available: http://www.cbc.ca/news/politics/story/2012/03/27/pol-ndp-voting-disruption-deliberate.html

[98] C. Culnane, M. Eldridge, A. Essex, and V. Teague, "Trust Implications of DDoS Protection in Online Elections," *arXiv preprint arXiv:1708.00991*, 2017.

[99] Intelligence Community Assessment. (2017, January) Assessing Russian Activities and Intentions in Recent US Elections. Office of the Drector of National Intelligence: National Intelligence Council. Accessed: September 2017. [Online]. Available: https://www.dni.gov/files/documents/ICA_2017_01.pdf

[100] BBC. (2017, January) Brexit vote site may have been attacked, MPs say in report. bbc. Accessed: September 2017. [Online]. Available: http://www.bbc.co.uk/news/uk-politics-39564289

[101] D. Pauli. (2014, June) 'Most sophisticated DDoS' ever strikes Hong Kong democracy poll. The Register. Accessed: September 2017. [Online]. Available: http://www.theregister.co.uk/2014/06/23/most_sophisticated_ddos_strikes_hk_democracy_poll/

[102] P. Penar. (2016, October) African citizens have very low levels of trust in how elections are run. [Online]. Available: https://theconversation.com/african-citizens-have-very-low-levels-of-trust-in-how-elections-are-run-66447

[103] Human Rights Watch. (2008, March) Ballots to Bullets: Organized Political Violence and Kenya's Crisis of Governance. Accessed: September 2017. [Online]. Available: https://www.hrw.org/report/2008/03/16/ballots-bullets/organized-political-violence-and-kenyas-crisis-governance

[104] The Economist. (2011, December) Voting, Russian-style. The Economist Newspaper Limited. Accessed: September 2017. [Online]. Available: http://www.economist.com/node/21541455

[105] F. Baiardi, A. Falleni, R. Granchi, F. Martinelli, M. Petrocchi, and A. Vaccarelli, "SEAS, a secure e-voting protocol: design and implementation," *Computers & Security*, vol. 24, no. 8, pp. 642–652, 2005.

[106] S. Campanelli, A. Falleni, F. Martinelli, M. Petrocchi, and A. Vaccarelli, "Mobile implementation and formal verification of an e-voting system." *ICIW*, vol. 8, pp. 476–481, 2008.

[107] Scytl. (2017) Scytl Phone Voting. Scytl Secure Electronic Voting, S.A. Accessed: September 2017. [Online]. Available: https://www.scytl.com/en/products/election-day/scytl-phone-voting/

[108] S. Dzieduszycka-Suinat, J. Murray, J. R. Kiniry, D. M. Zimmerman, D. Wagner, P. Robinson, A. Foltzer, and S. Morina. (2015, July) The Future Of Voting End-to-end verifiable Internet voting: Specification and Feasibility assessment study. US Vote Foundation. Accessed: September 2017. [Online]. Available: https://www.usvotefoundation.org/sites/default/files/E2EVIV_full_report.pdf

[109] D. Galindo and J. Puiggali. (2015, July) Response to the inaccuracies in the report by the US Vote Foundation and Galois Inc. Accessed: September 2017. [Online]. Available: https://www.scytl.com/wp-content/uploads/2015/07/Response-to-the-inaccuracies-in-the-US-Vote-Foundation-report-july-2015.pdf

[110] J. Cucurull, S. Guasch, and D. Galindo, "Transitioning to a Javascript Voting Client for Remote Online Voting," in *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016)*, 2016, pp. 121–132.

[111] J. A. Halderman and V. Teague, "The New South Wales iVote system: Security failures and verification flaws in a live online election," in *International Conference on E-Voting and Identity*. Springer, 2015, pp. 35–53.

[112] US-CERT. (2015, March) FREAK SSL/TLS Vulnerability. Homeland Security. Accessed: September 2017. [Online]. Available: https://www.us-cert.gov/ncas/current-activity/2015/03/06/FREAK-SSLTLS-Vulnerability

[113] Apple Inc. (2017) ApplePay. Accessed: September 2017. [Online]. Available: https://www.apple.com/uk/apple-pay/

[114] PayPal. (2017) PayPal Mobile App. Accessed: September 2017. [Online]. Available: https://www.paypal.com/uk/webapps/mpp/mobile-apps

[115] 3GPP. (1999) 3GPP TS 03.90 Unstructured Supplementary Service Data (USSD). Accessed: September 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=184

[116] ——. (1999) 3GPP TS 02.90 Unstructured Supplementary Service Data (USSD); Stage 1. Accessed: September 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=96

[117] P. Traynor, P. McDaniel, and T. La Porta, *Security for telecommunications networks*. Springer Science & Business Media, 2008, vol. 40.

[118] B. W. Nyamtiga, A. Sam, and L. S. Laizer, "Security Perspectives for USSD versus SMS in conducting mobile transactions: A case study of Tanzania," *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 1, no. 3, pp. 38–43, 2013.

[119] Coinapult. (2017) Coinapult SMS. Accessed: September 2017. [Online]. Available: https://coinapult.com/sms/info

[120] Gautham. (2016, September) Send Bitcoin Like an SMS on the New BTC.com Wallet App. Accessed: September 2017. [Online]. Available: http://www.newsbtc.com/2016/09/19/btc-com-wallet-app-sms-bitcoin/

[121] Coinbase. (2017, March) Coinbase. Accessed February 2017: Note - SMS service removed from website March 2017. [Online]. Available: https://www.coinbase.com/

[122] Bitwala. (2017, March) Send mobile money to Nigeria, Tanzania and Uganda for free with Bitwala. Accessed: September 2017. [Online]. Available: https://www.bitwala.io/send-bitcoin-to-mobile-money-for-free-africa

[123] J. Redman. (2017, March) Bitwala Connects Bitcoin to M-Pesa in Sub-Saharan Africa. Accessed: September 2017. [Online]. Available: https://news.bitcoin.com/bitwala-connects-bitcoin-mpesa-sub-saharan-africa/

[124] Coindesk. (2015, August) Bitcoin remittance startup 37Coins announces closure. Accessed: September 2017. [Online]. Available: http://www.coindesk.com/bitcoin-remittance-startup-37coins-announces-closure/

[125] J. Young. (2016, January) Former Kipochi CTO Explains Controversial M-Pesa Deal. Accessed: September 2017. [Online]. Available: http://www.newsbtc.com/2016/01/11/former-kipochi-ceo-explains-controversial-m-pesa-bitcoin-deal/

[126] Bitpesa. (2015, December) Bitpesa v Safaricom. Accessed: September 2017. [Online]. Available: https://www.bitpesa.co/blog/bitpesa-v-safaricom/

[127] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, and J. P. Richer. (2016, December) DRAFT NIST Special Publication 800-63B, Digital Authentication Guideline: Authentication and Lifecycle Management. National Institute of Standards and Technology (NIST). [Online]. Available: https://pages.nist.gov/800-63-3/sp800-63b.html

[128] M. Paik, "Stragglers of the herd get eaten: security concerns for GSM mobile banking applications," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*. ACM, 2010, pp. 54–59.

[129] FlexiSpy. (2017) FlexiSpy: Spoof SMS. Accessed: September 2017. [Online]. Available: https://www.flexispy.com/en/features/spoof-sms.htm

[130] Telco 2.0. (2010, February) Security Breach at M-PESA: Telco 2.0 Crash Investigation. Accessed: September 2017. [Online]. Available: http://www.telco2.net/blog/2010/02/security_breach_at_mpesa_telco.html

[131] J. Bickford, R. O'Hare, A. Baliga, V. Ganapathy, and L. Iftode, "Rootkits on smart phones: attacks, implications and opportunities," in *Proceedings of the*

Eleventh Workshop on Mobile Computing Systems & Applications. ACM, 2010, pp. 49–54.

[132] D. Goodin. (2011, February) ZeuS trojan attacks bank's 2-factor authentication. Accessed: September 2017. [Online]. Available: http://http://www.theregister.co.uk/2011/02/22/zeus_2_factor_authentication_attack/

[133] C. Mulliner, N. Golde, and J.-P. Seifert, "SMS of Death: From Analyzing to Attacking Mobile Phones on a Large Scale," in *USENIX Security Symposium*, 2011.

[134] S. Gibbs. (2016, April) SS7 hack explained: what can you do about it? . The Guardian. Accessed: September 2017. [Online]. Available: https://www.theguardian.com/technology/2016/apr/19/ss7-hack-explained-mobile-phone-vulnerability-snooping-texts-calls

[135] OWASP. (2013) Open Web Application Security Project (OWASP): Top Ten Project. The Open Web Application Security Project. Accessed: September 2017. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[136] J. Leyden. (2012, October) HSBC websites fell in DDoS attack last night, bank admits. Accessed: September 2017. [Online]. Available: http://www.theregister.co.uk/2012/10/19/hsbc_ddos/

[137] Trend Micro. (2018, February) 2017 Mobile Threat Landscape. Accessed March 2018. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/research-and-analysis/threat-reports/roundup/2017-mobile-threat-landscape

[138] G. D. R. Samani. (2018) McAfee Mobile Threat Report 2018. Accessed March 2018. [Online]. Available: https://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2018.pdf

[139] F-Secure. (2017) Worm: SymbOS/Commwarrior Threat Description. F-Secure. Accessed: September 2017. [Online]. Available: http://www.f-secure.com/v-descs/commwarrior.shtml

[140] ISO/IEC. ISO/IEC 18092 Information Technology – telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1). Accessed: September 2017. [Online]. Available: http://www.iso.org

[141] ——. ISO/IEC 14443 Identification cards – Contactless integrated circuit cards – Proximity cards . Accessed: September 2017. [Online]. Available: http://www.iso.org

[142] ——. ISO/IEC 15693 Identification cards – Contactless integrated circuit cards – Vicinity cards. Accessed: September 2017. [Online]. Available: http://www.iso.org

[143] ——. (2012) ISO/IEC 21481:2012 Information technology – Telecommunications and information exchange between systems – Near Field Communication Interface and Protocol -2 (NFCIP-2). Accessed: September 2017. [Online]. Available: https://www.iso.org/standard/56855.html

[144] Sony. (2017) What is Felica? Accessed: September 2017. [Online]. Available: http://www.sony.net/Products/felica/about/

[145] Blizzard Entertainment Inc. (2017) World of Warcraft. Accessed: September 2017. [Online]. Available: http://eu.battle.net/wow/en/

[146] Y. Liu, J. Yang, and M. Liu, "Recognition of QR code with mobile phones," in *Control and Decision Conference, 2008. CCDC 2008. Chinese.* IEEE, 2008, pp. 203–206.

[147] P. Kieseberg, M. Leithner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, and E. Weippl, "QR code security," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia.* ACM, 2010, pp. 430–435.

[148] R. Boden. (2015, August) Indian banks to test mVisa mobile payments in Bangalore. Accessed: September 2017. [Online]. Available: https://www.nfcworld.com/2015/08/06/336976/indian-banks-to-test-mvisa-mobile-payments-in-bangalore/

[149] S. Atkins. (2017, March) mVisa to expand to 10 countries. ContactlessIntelligence. Accessed: September 2017. [Online]. Available: https://contactlessintelligence.com/2017/03/01/mvisa-to-expand-to-10-countries/

[150] J. Z. Gao, L. Prakash, and R. Jagatesan, "Understanding 2D-barcode technology and applications in m-commerce-design and implementation of a 2D barcode processing solution," in *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, vol. 2. IEEE, 2007, pp. 49–56.

[151] T. Parikh, P. Javid, K. Ghosh, K. Toyama *et al.*, "Mobile phones and paper documents: evaluating a new approach for capturing microfinance data in rural India," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 551–560.

[152] J. Hart, K. Markantonakis, and K. Mayes, "Website credential storage and two-factor web authentication with a Java SIM," *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*, pp. 229–236, 2010.

[153] G. Alpár, L. Batina, and R. Verdult, "Using NFC phones for proving credentials," in *International GI/ITG Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*. Springer, 2012, pp. 317–330.

[154] Chirp. What Is Chirp? Asio Ltd. Accessed: September 2017. [Online]. Available: https://www.chirp.io/#what-is-chirp

[155] Skylanders. (2017) Skylanders imaginators. Activision Publishing. Inc. Accessed: September 2017. [Online]. Available: https://www.skylanders.com/uk/en/video-games/skylanders-imaginators

[156] TagPay. (2017) Tagpay. Accessed: September 2017. [Online]. Available: http://en.tagpay.fr/

[157] (2017) About Aadhaar. Unique Identitification Authority of India. [Online]. Available: https://uidai.gov.in/your-aadhaar/about-aadhaar.html

[158] G. Kambourakis, D. Damopoulos, D. Papamartzivanos, and E. Pavlidakis, "Introducing touchstroke: keystroke-based authentication system for smartphones," *Security and Communication Networks*, 2014.

[159] F. Alshanketi, I. Traore, and A. A. Ahmed, "Improving Performance and Usability in Mobile Keystroke Dynamic Biometric Authentication," in *Security and Privacy Workshops (SPW), 2016 IEEE*. IEEE, 2016, pp. 66–73.

[160] C. Raes. (2016, October) Mastercard makes fingerprint and 'selfie' payment technology a reality. MasterCard. Accessed: September 2017. [Online]. Available: http://newsroom.mastercard.com/eu/press-releases/mastercard-makes-fingerprint-and-selfie-payment-technology-a-reality/

[161] Biometric Update.com. (2017, March) Entrust Datacard building out continuous authentication capabilities. Biometrics Research Group, Inc. Accessed: September 2017. [Online]. Available: https://www.biometricupdate.com/201702/entrust-datacard-building-out-continuous-authentication-capabilities

[162] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[163] A. Vapen, D. Byers, and N. Shahmehri, "2-clickAuth Optical Challenge-Response Authentication," in *Availability, Reliability, and Security, 2010. ARES'10.* IEEE, 2010, pp. pp. 79–86.

[164] I. Jorstad, T. Jonvik *et al.*, "Strong authentication with mobile phone as security token," in *Mobile Adhoc and Sensor Systems, 2009. MASS'09.* IEEE, 2009, pp. 777–782.

[165] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Security and Privacy (SP), 2012 IEEE Symposium on.* IEEE, 2012, pp. 553–567.

[166] B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," in *Financial Cryptography*, vol. 4107. Springer, 2006, pp. 1–19.

[167] Vasco. (2017) CRONTO Visual Transaction Signing . Accessed: September 2017. [Online]. Available: https://www.vasco.com/products/two-factor-authenticators/crontosign.html

[168] M. Mannan and P. C. van Oorschot, "Leveraging personal devices for stronger password authentication from untrusted computers," *Journal of Computer Security*, vol. 19, no. 4, pp. 703–750, 2011.

[169] C. Morningstar and F. Farmer, "The lessons of Lucasfilm's habitat," in *Cyberspace.* MIT Press, 1991, pp. 273–302.

[170] M. Bell, "Virtual Worlds Research: Past, Present & Future July 2008: Toward a definition of "Virtual Worlds"," 2008.

[171] R. Reynolds. (2005, August) The four worlds theory. Terra Nova. Accessed: September 2017. [Online]. Available: http://terranova.blogs.com/terra_nova/2005/08/the_four_worlds.html

[172] Linden Research Inc. Second Life Official Website. Linden Research Inc. Accessed: September 2017. [Online]. Available: http://secondlife.com/

[173] R. Schroeder, "Defining virtual worlds and virtual environments," *Journal of Virtual Worlds Research*, vol. 1, no. 1, pp. 2–3, 2008.

[174] Kaneva. (2017) Kaneva. Accessed: September 2017. [Online]. Available: http://www.kaneva.com

[175] Outsmart. (2017) Smallworlds. Outsmart. Accessed: September 2017. [Online]. Available: https://www.smallworlds.com/

[176] M. Korolov. (2015, November) Second Life GDP totals $500 Million. Accessed March 2018. [Online]. Available: http://www.hypergridbusiness.com/2015/11/second-life-gdp-totals-500-million/

[177] K. Watkins. (2015, May) Ebbe: SL users cashed out $60mil last year. Accessed March 2018. [Online]. Available: http://www.hypergridbusiness.com/2015/05/ebbe-sl-users-cashed-out-60-mil-last-year/

[178] New World Notes. (2017, April) Second Life Content Creators Now Likely Making More Money from Second Life Than Second Life's Own Corporate Owner. Accessed March 2018. [Online]. Available: http://nwn.blogs.com/nwn/2017/04/second-life-sl-marketplace-linden-lab-sl-revenue.html

[179] C. Lee and M. Warren, "Security issues within Virtual Worlds such as Second Life," in *Australian Information Security Management Conference*, 2007, p. 44.

[180] C. Lee, "Understanding security threats in Virtual Worlds," in *Americas Conference on Information Systems*, 2009.

[181] G. Hogben, D. Barroso, R. Bartle, P. Chazerand, M. de Zwart, J. Doumen, S. Gorniak, E. Guomundsson, M. Kazmierczak, M. Kaskenmaa, D. B. Lopez, A. Martin, I. Naumann, R. Reynolds, J. Richardson, C. Rossow, A. Rywczynska, and M. Thumann. (2008, November) Position Paper: Virtual Worlds, Real Money. European Union Agency for Network and Information Security (ENISA). Accessed: September 2017. [Online]. Available: http://www.enisa.europa.eu/publications/archive/security-and-privacy-in-virtual-worlds-and-gaming

[182] European Parliament, Council. (1995, October) European Privacy Directive 95/46. Accessed: September 2017. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:NOT

[183] M. Sparkes. (2013, December) NSA and GCHQ spies 'operated in games including World of Warcraft and Second Life'. The Telegraph. Accessed: September 2017. [Online]. Available: http://www.telegraph.co.uk/technology/video-games/video-game-news/10506017/NSA-and-GCHQ-spies-operated-in-games-including-World-of-Warcraft-and-Second-Life.html

[184] M. Mazzeti and J. Elliott. (2013, December) Spies Infiltrate a Fantasy Realm of Online Games. Accessed: September 2017. [Online]. Available: http://www.nytimes.com/2013/12/10/world/spies-dragnet-reaches-a-playing-field-of-elves-and-trolls.html

[185] Eve Online. (2017) Eve Online. Accessed: September 2017. [Online]. Available: https://www.eveonline.com/

[186] CCP t20. (2007) On Recent Allegations. Eve Online. Accessed: September 2017. [Online]. Available: https://community.eveonline.com/news/dev-blogs/on-recent-allegations/

[187] John R. (2016, March) Outsourcing Fun: Gold Farming & the Rise of Digital Sweatshops. Accessed March 2018. [Online]. Available: https://onlineeconomy.hbs.org/submission/outsourcing-fun-gold-farming-the-rise-of-digital-sweatshops/

[188] D. Winder. (2013, August) When Gaming Attacks Get Serious. Alphr. Accessed: September 2017. [Online]. Available: http://www.alphr.com/realworld/383737/when-gaming-attacks-get-serious

[189] P. Pollack. (2006, August) Online banker runs off with cash, avatars cry foul. Arstechnica. Accessed: September 2017. [Online]. Available: http://arstechnica.com/uncategorized/2006/08/7605/

[190] S. Spring. (2006, December) Games: Virtual thievery. Newsweek. Accessed: September 2017. [Online]. Available: http://www.newsweek.com/games-virtual-thievery-105691

[191] E. Cavalli. (2008, February) Police refuse to aid in virtual theft case. Wired. Accessed: September 2017. [Online]. Available: http://www.wired.com/gamelife/2008/02/police-refuse-t/

[192] D. Goodin. (2014, January) World of Warcraft users hit by account-hijacking malware attack. Arstechnica. Accessed: September 2017. [Online]. Available: http://arstechnica.com/security/2014/01/world-of-warcraft-users-hit-by-account-hijacking-malware-attack/

[193] G. McGraw and M. Chow, "Guest Editors' Introduction: Securing Online Games: Safeguarding the Future of Software Security: How world of Warcraft Almost Ruined My credit rating," *Security & Privacy, IEEE*, vol. 7, no. 3, pp. 11–12, 2009. [Online]. Available: http://www2.computer.org/cms/Computer.org/dl/mags/sp/2009/03/extras/msp2009030011s.pdf

[194] K. Choo and R. Smith, "Criminal exploitation of online systems by organised crime groups," *Asian journal of criminology*, pp. pp. 37–59, 2008.

[195] R. Stokes, "Virtual money laundering: the case of bitcoin and the linden dollar," *Information & Communications Technology Law*, vol. 21, no. 3, pp. 221–236, 2012. [Online]. Available: https://doi.org/10.1080/13600834.2012.744225

[196] N. Yee, "The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments," *Presence: Teleoperators and virtual environments*, vol. 15, no. 3, pp. 309–329, 2006.

[197] I. Muttick, "Securing Virtual Worlds Against Real Attacks -The challenges of online game development," McAfee, Inc, Tech. Rep., 2008. [Online]. Available: https://www.info-point-security.com/open_downloads/2008/McAfee_wp_online_gaming_0808.pdf

[198] Second Life Community. (2017) Second Life Forums. Linden Labs. Accessed: September 2017. [Online]. Available: https://community.secondlife.com/forums/

[199] mmorpg.com. (2017) MMORPG Discussion Forums. Accessed: September 2017. [Online]. Available: http://forums.mmorpg.com

[200] Blizzard Entertainment Inc. (2017) Blizzard shop. Accessed: September 2017. [Online]. Available: https://eu.battle.net/shop/en/

[201] Eve Online. (2017) Eve Online Marketplace. Eve Online. Accessed: September 2017. [Online]. Available: https://forums.eveonline.com/default.aspx?g=forum&c=67

[202] Linden Research Inc. (2013) Avaline. Linden Research Inc. Accessed: September 2017. [Online]. Available: http://www.kzero.co.uk/blog/second-life-rings-the-changes-with-avaline/

[203] Linden Labs. (2009, May) Over 15 Billion Minutes of Voice Have Been Delivered in Second Life. Linden Labs Inc. Accessed: September 2017. [Online]. Available: https://www.lindenlab.com/releases/over-15-billion-minutes-of-voice-have-been-delivered-in-second-life

[204] Blizzard Entertainment Inc. (2017) Battle.net SMS Protect. Blizzard Entertainment Inc. Accessed: September 2017. [Online]. Available: https://eu.battle.net/support/en/article/26824

[205] Gamepedia. (2013, August) Battle.net Mobile Authenticator. Gamepedia. Accessed: September 2017. [Online]. Available: https://wow.gamepedia.com/Battle.net_Mobile_Authenticator#cite_note-3

[206] bugtraq. (2010) Battle.net mobile authenticator mitm vulnerability. seclists.org. Accessed: September 2017. [Online]. Available: http://seclists.org/bugtraq/2010/Sep/160

[207] Just Ask Gemalto. (2017) What is a UICC and how is it different from a SIM card? Gemalto. Accessed: September 2017. [Online]. Available: https://www.justaskgemalto.com/en/what-uicc-and-how-it-different-sim-card/

[208] GSMA Intelligence. (2015, March) Understanding SIM evolution. Accessed: September 2017. [Online]. Available: https://www.gsmaintelligence.com/research/2015/03/understanding-sim-evolution/499/

[209] GSMA. (2017) Mobile Technology. Accessed: September 2017. [Online]. Available: https://www.gsma.com/aboutus/gsm-technology

[210] B. Kaliski and J. Staddon, "PKCS# 1: RSA cryptography specifications version 2.0," RFC 2437, October, Tech. Rep., 1998.

[211] G. Locke and P. Gallagher, "FIPS PUB 186-3: Digital signature standard (DSS)," *Federal Information Processing Standards Publication*, 2009.

[212] Safaricom. (2017) M-PESA. Accessed: September 2017. [Online]. Available: http://www.safaricom.co.ke/index.php?id=250

[213] ETSI. (2017) SCP Group : SCP Specifications. ETSI SCP Group. Accessed: September 2017. [Online]. Available: https://portal.etsi.org/tb.aspx?tbid=534&SubTB=534,639,640,714

[214] 3GPP. (2017) 3rd Generation Partnership Project (3GPP) . Accessed: September 2017. [Online]. Available: http://www.3gpp.org/

[215] ISO/IEC. (2011, February) ISO/IEC 7816-1:2011: Identification cards – Integrated circuit cards – Part 1: Cards with contacts – Physical characteristics. Accessed: September 2017. [Online]. Available: https://www.iso.org/standard/54089.html

[216] ——. (2007, October) ISO/IEC 7816-2:2007: Identification cards – Integrated circuit cards – Part 2: Cards with contacts – Dimensions and location of the contacts. Accessed: September 2017. [Online]. Available: https://www.iso.org/standard/45989.html

[217] ——. (2006, November) ISO/IEC 7816-3:2006 Identification cards – Integrated circuit cards – Part 3: Cards with contacts – Electrical interface and transmission protocols. Accessed: September 2017. [Online]. Available: https://www.iso.org/standard/38770.html

[218] ——. (2013, April) ISO/IEC 7816-4:2013 Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange. Accessed: September 2017. [Online]. Available: https://www.iso.org/standard/54550.html

[219] ISO. (2017) International Organization for Standardization. Accessed: September 2017. [Online]. Available: https://www.iso.org/home.html

[220] 3GPP. (2015) 3GPP TS 11.14 Specification of the SIM Application Toolkit (SAT) for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface. Accessed: September 2017. [Online]. Available: www.3gpp.org/DynaReport/1114.htm

[221] ——. (2015) 3GPP TS 31.111 Universal Subscriber Identity Module (USIM) Application Toolkit (USAT). Accessed: September 2017. [Online]. Available: www.3gpp.org/DynaReport/31111.htm

[222] ——. (1998) 3GPP TS 03.19 Subscriber Identity Module Application Programming Interface (SIM API) for Java Card. Accessed: September 2017. [Online]. Available: www.3gpp.org/DynaReport/0319.htm

[223] ——. (2015) 3GPP TS 31.130 SIM Application Programming Interface (API); SIM API for Java Card (TM). Accessed: September 2017. [Online]. Available: www.3gpp.org/DynaReport/31130.htm

[224] ——. (2015) 3GPP TS 03.48 Security mechanisms for SIM application Toolkit; Stage 2. Accessed: September 2017. [Online]. Available: www.3gpp.org/DynaReport/0348.htm

[225] GlobalPlatform. (2017) GlobalPlatform Card Specifications. Accessed: September 2017. [Online]. Available: http://www.globalplatform.org/specificationscard.asp

[226] OMA. (2017) Open Mobile Alliance. Accessed: September 2017. [Online]. Available: http://www.openmobilealliance.org/

[227] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Rfc 2616, hypertext transfer protocol–http/1.1, 1999," *URL http://www. rfc. net/rfc2616. html*, 2009.

[228] SimAlliance. (2009, February) Smart Card Web Server: How to bring operators' applications and services to the mass market. SIMAlliance. Accessed: September 2017. [Online]. Available: http://simalliance.org/wp-content/uploads/2015/03/WP_SIMallianceSCWS_Feb09_Final.pdf

[229] L. Kyrillidis, K. Mayes, and K. Markantonakis, "Web Server on a SIM card," *Lecture Notes in Engineering and Computer Science*, vol. 2183, 2010.

[230] J. Franks. (1999) HTTP Authentication: Basic and Digest Access Authentication. Accessed: September 2017. [Online]. Available: http://www.ietf.org/rfc/rfc2617.txt

[231] P. Eronen and H. Tschofenig, "RFC4279: Pre-shared key ciphersuites for transport layer security (TLS)," *Internet Engineering Task Force*, 2005.

[232] NIST. (2001, November) Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197. Accessed: September 2017. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[233] Internet Engineering Task Force (IET). (2000, May) RFC 2818: Hypertext Transfer Protocol over TLS protocol, May 2000. Internet Engineering Task Force (IET). Accessed: September 2017. [Online]. Available: http://www.ietf.org/rfc/rfc2818.txt

[234] ETSI. (2016) TS 102 223 V13.1.0 (2016-09) Smart Cards; Card Application Toolkit (CAT) (Release 13) . Accessed: September 2017. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102200_102299/102223/13.01.00_60/ts_102223v130100p.pdf

[235] Oracle. (2017) Java Card Connected Platform Specification v3.0.1. Oracle. Accessed: September 2017. [Online]. Available: http://www.oracle.com/technetwork/java/javame/javacard/download/default-1492179.html

[236] Open Mobile Alliance. (2011, August) OMA SIP Push V1.0 . Accessed: September 2017. [Online]. Available: http://www.oma-works.org/Technical/release_program/SIP_Push_v1_0.aspx

[237] GlobalPlatform, "Remote Application Management over HTTP Card Specification v2.2 Amendment B Version 1.1.1," March 2012.

[238] GlobalPlatform. (2009, April) GlobalPlatform's Proposition for NFC Mobile: Secure Element Management and Messaging . GlobalPlatform Inc. Accessed: September 2017. [Online]. Available: http://www.globalplatform.org/documents/GlobalPlatform_NFC_Mobile_White_Paper.pdf

[239] SimAlliance. (2009, December) Smart Card Web Server Stepping Stones OMA 1.1 Version 1.0.0. Accessed: September 2017. [Online]. Available: http://simalliance.org/wp-content/uploads/2015/03/SCWS_SteppingStones_2009_v1.0.01.pdf

[240] Republic of Estonia: Information System Authority. (2017) Public Key Infrastructure PKI. Accessed: September 2017. [Online]. Available: https://ria.ee/en/public-key-infrastructure.html

[241] NIST. (2007, March) Recommendation for Key Management - Part 1: General (Revised).Special Publication 800-57. National Institute of Standards and Technology (NIST). Accessed: September 2017. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-57/

[242] SANS. (2017) Password Construction Guidelines. SANS. Accessed: September 2017. [Online]. Available: https://www.sans.org/security-resources/policies/general#password-construction-guidelines

[243] W. Chen, K. Mayes, Y. Lien, and J. Chiu, "NFC mobile payment with Citizen Digital Certificate," in *Next Generation Information Technology (ICNIT), 2011 The 2nd International Conference on.* IEEE, 2011, pp. 120–126.

[244] D. Chaum, P. Ryan, and S. Schneider, "A practical voter-verifiable election scheme," *Computer Security–ESORICS 2005*, pp. 118–139, 2005.

[245] P. Ryan and S. Schneider, "Prêt à Voter with re-encryption mixes," *Computer Security–ESORICS 2006*, pp. 313–326, 2006.

[246] P. Ryan, "Prêt à Voter with confirmation codes," in *Proceedings of the USENIX Electronic Voting Technology Workshop*, 2011.

[247] L. Zhou, F. Schneider, M. Marsh, and A. Redz, "Distributed blinding for distributed ElGamal re-encryption," in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on.* IEEE, 2005, pp. 824–824.

[248] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman, "Security analysis of the Estonian internet voting system," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security.* ACM, 2014, pp. 703–715.

[249] J. R. Nurse, I. Agrafiotis, A. Erola, M. Bada, T. Roberts, M. Williams, M. Goldsmith, and S. Creese, "An Assessment of the Security and Transparency Procedural Components of the Estonian Internet Voting System," in *International Conference on Human Aspects of Information Security, Privacy, and Trust.* Springer, 2017, pp. 366–383.

[250] S. Heiberg and J. Willemson, "Verifiable internet voting in Estonia," in *Electronic Voting: Verifying the Vote (EVOTE), 2014 6th International Conference on.* IEEE, 2014, pp. 1–8.

[251] S. Heiberg, T. Martens, P. Vinkel, and J. Willemson, "Improving the verifiability of the Estonian Internet Voting scheme," in *International Joint Conference on Electronic Voting.* Springer, 2016, pp. 92–107.

[252] P. Laud and M. Roos, "Formal analysis of the Estonian Mobile-id protocol," in *Nordic Conference on Secure IT Systems.* Springer, 2009, pp. 271–286.

[253] M. Clarkson, S. Chong, and A. Myers, "Civitas: Toward a secure voting system," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on.* IEEE, 2008, pp. 354–368.

[254] J. Clark and U. Hengartner, "Panic Passwords: Authenticating under Duress." *HotSec*, vol. 8, p. 8, 2008.

[255] S. Shinde and A. Lele. (2012, September) mChek might have suspended operations. Business Standard. Accessed: September 2017. [Online]. Available: http://www.business-standard.com/article/finance/mchek-might-have-suspended-operations-sources-112092700096_1.html

[256] H. Thinyane and M. Thinyane, "ICANSEE: A SIM based application for digital inclusion of the visually impaired community," in *Innovations for Digital Inclusions, 2009. K-IDI 2009. ITU-T Kaleidoscope:.* IEEE, 2009, pp. 1–6.

[257] GlobalPlatform, "Confidential Card Content Management GlobalPlatform Card Specification v2.2 - Amendment A v1.0.1," January 2011.

[258] I. Mas and H. Siedek. (2008, May) Banking through networks of retail agents. CGAP. Accessed: September 2017. [Online]. Available: http://www.cgap.org/publications/banking-through-networks-retail-agents

[259] Bitcoin.org. (2017) Bitcoin Wiki. Accessed: September 2017. [Online]. Available: https://en.bitcoin.it/wiki/Main_Page

[260] S. Nakamoto, "Bitcoin: A peer-to-peer e-cash system," *Bitcoin.org*, 2008. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[261] CAF. (2015, December) Giving Unchained: Philanthropy and the Blockchain. Charities Aid Foundation. Accessed: September 2017. [Online]. Available: https://www.cafonline.org/docs/default-source/about-us-publications/givingunchained-philanthropy-and-the-blockchain.pdf?sfvrsn=4

[262] Y. B. Perez. (2015, September) Can Bitcoin Make a Difference in the Global Aid Sector? Coindesk. Accessed: September 2017. [Online]. Available: http://www.coindesk.com/can-bitcoin-make-a-difference-in-the-global-aid-sector/

[263] R. Davies, *Public Good by Private Means: How philanthropy shapes Britain.* Alliance Publishing Trust, 2016.

[264] ImpACT Coalition. (2013) Through a glass DARKLY: The case for accelerating the drive for accountability, clarity and transparency in the charity sector. Accessed: September 2017. [Online]. Available: http://www.cfg.org.uk/news/press-releases/2013/june/~/media/Files/Resources/Briefings/Through%20a%20Glass%20Darkly.ashx

[265] S. Birkwood. (2015, January) Is Bitcoin the ideal charity currency or a cause for concern? Third Sector. Accessed: September 2017. [Online]. Available: http://www.thirdsector.co.uk/analysis-bitcoin-ideal-charity-currency-cause-concern/fundraising/article/1326549

[266] BitGive. (2017, March) Givetrack: Donation tracking. BitGive Foundatiom. Accessed: September 2017. [Online]. Available: https://www.bitgivefoundation.org/givetrack_/

[267] Bitgo. (2017) Bitgo. Accessed: September 2017. [Online]. Available: https://www.bitgo.com/

[268] BBC. (2017, June) Accenture and Microsoft plan digital IDs for millions of refugees. Accessed: September 2017. [Online]. Available: http://www.bbc.co.uk/news/technology-40341511

[269] RSK. (2017) Rootstock Platform. RSK Labs. Accessed: September 2017. [Online]. Available: http://www.rsk.co/

[270] D. Gilman, "Cyber-warfare and humanitarian space," in *Communications Technology and Humanitarian Delivery Challenges and Opportunities for Security Risk Management, European Interagency Security Forum (EISF)*, 2014, accessed: September 2017. [Online]. Available: http://commstech-hub.eisf.eu/uploads/4/0/2/4/40242315/daniel_gilman_cyberwarfare_and_humanitarian_space_eisf_october_2014.pdf

[271] J. Rydell, D. M'Raihi, M. Pei, and S. Machani. (2011, May) TOTP: Time-Based One-Time Password Algorithm. Accessed: September 2017. [Online]. Available: https://tools.ietf.org/html/rfc6238

[272] N. Haller, C. Metz, P. Nesser, and M. Straw, "A One Time Password system RFC 2289," *Internet Engineering Task Force.*, 1998.

[273] The Charity Commission, "Compliance Toolkit Protecting Charities from Harm: Due Diligence, Monitoring and Verification of End Use of Charitable Funds," Tech. Rep., Visited, December 2016. [Online]. Available: http://forms.charitycommission.gov.uk/media/89350/compliance_toolkit_2.pdf

[274] T. Dierks and E. Rescorla. (2008, August) The Transport Layer Security (TLS) Protocol Version 1.2. IETF. Accessed: September 2017. [Online]. Available: https://tools.ietf.org/html/rfc5246

[275] ETSI. TS 102 600 V7.5.0 (2009-04): Technical Specification: Smart Cards; UICC- Terminal interface; Characteristics of the USB interface (Release 7). ETSI. Accessed: September 2017. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102600_102699/102600/07.05.00_60/ts_102600v070500p.pdf

[276] Bluetooth SIG Inc. (2017) What is Bluetooth? - How it Works. Accessed: September 2017. [Online]. Available: https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works

[277] Wi-Fi Alliance. (2017) Wi-Fi Direct. Accessed: September 2017. [Online]. Available: http://www.wi-fi.org/discover-wi-fi/wi-fi-direct

[278] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. (2008, May) Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF. Accessed: September 2017. [Online]. Available: https://tools.ietf.org/html/rfc5280

[279] imei.info. (2016) What is ICCID? Accessed: September 2017. [Online]. Available: http://www.imei.info/faq-what-is-ICCID/

[280] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.

[281] M. H. Hassoun, *Fundamentals of artificial neural networks*. MIT press, 1995.

[282] E. Keogh and C. A. Ratanamahatana, "Exact indexing of Dynamic Time Warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[283] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[284] BBC. (2015, April) Intel RealSense 3D depth camera fitted into smartphone. Accessed: September 2017. [Online]. Available: http://www.bbc.co.uk/news/technology-32215577

[285] Google. (2017) Project Tango. Accessed: September 2017. [Online]. Available: https://get.google.com/tango/

[286] Asus. (2017) ZenFone AR (ZS571KL). Asus. Accessed: September 2017. [Online]. Available: https://www.asus.com/Phone/ZenFone-AR-ZS571KL/

[287] S. J. Grunewald. (2016, February) Depth-Sensing Cameras Will Soon Turn Every Smartphone into a High-Quality 3D Scanner. 3dPrint.com. Accessed: September 2017. [Online]. Available: https://3dprint.com/117809/depth-sensing-phone-cameras/

[288] A. Shah. (2016, January) Intel's smartphone with integrated RealSense 3D camera to ship for $399. PC World. Accessed: September 2017. [Online]. Available: http://www.pcworld.com/article/3019937/hardware/intels-smartphone-with-integrated-realsense-3d-camera-to-ship-for-399.html

[289] Lenovo. (2017, June) Lenovo Phab 2 Pro. Accessed: September 2017. [Online]. Available: http://www3.lenovo.com/us/en/smart-devices/-lenovo-smartphones/phab-series/Lenovo-Phab-2-Pro/p/WMD00000220

[290] Infineon. (2016, June) Smallest 3D camera worldwide brings Augmented Reality to a smartphone. Accessed: September 2017. [Online]. Available: http://www.infineon.com/cms/en/about-infineon/press/press-releases/2016/INFXX201606-064.html

[291] A. Shilov. (2016, January) Intel and Google Equip Smartphones with 3D Cameras and Computer Vision. AnandTech. Accessed: September 2017. [Online]. Available: http://www.anandtech.com/show/9940/intel-and-google-equip-smartphones-with-3d-cameras-and-computer-vision

[292] Leap Motion Inc. (2017) Leap Motion Developer. Accessed: September 2017. [Online]. Available: https://developer.leapmotion.com

[293] D. Holz. (2016, December) Leap Motion Goes Mobile. Leap Motion Blog. Accessed: September 2017. [Online]. Available: http://blog.leapmotion.com/mobile-platform/

[294] Microsoft, "Kinect for Windows features," http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx, 2013, [Online; accessed 17 September 2013].

[295] M. T. I. Aumi and S. Kratz, "AirAuth: evaluating in-air hand gestures for authentication," in *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, 2014, pp. 309–318.

[296] J. Wu, J. Konrad, and P. Ishwar, "Dynamic time warping for gesture-based user identification and authentication with Kinect," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2371–2375.

[297] J. Wu, P. Ishwar, and J. Konrad, "Silhouettes versus skeletons in gesture-based authentication with Kinect," in *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*. IEEE, 2014, pp. 99–106.

[298] J. Tian, C. Qu, W. Xu, and S. Wang, "Kinwrite: Handwriting-based authentication using kinect." in *NDSS*, 2013.

[299] CCP Games. (2013) The Council of Stellar Management. CCP Games - Eve Online. Accessed: September 2017. [Online]. Available: http://community.eveonline.com/community/csm/

[300] Battle.net Social Community. (2017) Real ID. Blizzard Entertainment Inc. Accessed: September 2017. [Online]. Available: http://us.battle.net/en/realid/

[301] Sachin Shetty, Symantec. (2010, November) Introduction to Spyware Keyloggers. http://www.symantec.com/connect/articles/introduction-spyware-keyloggers.

[302] SANS Institute InfoSec Reading Room. (2000, November) Introduction to IP Spoofing. SANS. Accessed: September 2017. [Online]. Available: https://uk.sans.org/reading-room/whitepapers/threats/introduction-ip-spoofing-959

[303] EntropiaPlanetsWiki. (2011, November) Entropia Universe takes Virtual World Citizenship to a whole new level! Mindark. Accessed: September 2017. [Online]. Available: http://www.entropiaplanets.com/wiki/Calypso_Land_Deeds

[304] Fandom. (2011) Voting stations. Fandom. Accessed: September 2017. [Online]. Available: http://secondlife.wikia.com/wiki/Voting_Stations

[305] S. Fernandes, R. Antonello, J. Moreira, D. Sadok, and C. Kamienski, "Traffic analysis beyond this world: the case of second life," in *Proceedings of the 17th International workshop on Network and Operating Systems Support for Digital Audio & Video (NOSSDAV)*. Citeseer, 2007.

[306] L. Ahn, M. Blum, N. Hopper, and J. Langford, "CAPTCHA: Using Hard AI Problems for Security," in *Advances in Cryptology, EUROCRYPT 2003*, ser. Lecture Notes in Computer Science, E. Biham, Ed. Springer Berlin Heidelberg, 2003, vol. 2656, pp. 294–311, accessed: September 2017. [Online]. Available: http://dx.doi.org/10.1007/3-540-39200-9_18

[307] NIST. (2015) NIST Policy on Hash Functions. Accessed: September 2017. [Online]. Available: http://csrc.nist.gov/groups/ST/hash/policy.html

[308] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *Information Theory, IEEE Transactions on*, vol. 31, no. 4, pp. 469–472, 1985.

[309] J. R. Douceur, "The Sybil Attack," in *Peer-to-peer Systems*. Springer, 2002, pp. 251–260.

[310] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni *et al.*, "DROWN: Breaking TLS using SSLv2," *Proceedings of the 25th USENIX Security Symposium*, August 2016.

[311] D. Goodin. (2012, September) Crack in Internet's foundation of trust allows HTTPS session hijacking. Arstechnica. Accessed: Septem-

ber 2017. [Online]. Available: http://arstechnica.com/security/2012/09/crime-hijacks-https-sessions/

[312] miTLS. (2015) A Zoo of TLS attacks. Microsoft-Inria Joint Centre. Accessed: September 2017. [Online]. Available: https://mitls.org/pages/attacks

[313] Common Criteria. (2017) Common Criteria. Accessed: September 2017. [Online]. Available: http://www.commoncriteriaportal.org/

[314] SFR S.A. (2010) (U)SIM Java Card Platform Protection Profile: Basic and SCWS Configurations. Accessed: September 2017. [Online]. Available: http://www.commoncriteriaportal.org/files/ppfiles/anssi-cc-cible_pp-2010-04en.pdf

[315] OWASP Top 10 Application Security Risks - 2017. Accessed March 2018. [Online]. Available: https://www.owasp.org/index.php/Top_10-2017_Top_10

[316] E. Vètillard. (2009) New Security Issues related to Embedded Web Servers . Accessed: September 2017. [Online]. Available: https://www.slideshare.net/EricVtillard/new-security-issues-related-to-embedded-web-servers

[317] e-Estonia. (2017) e-Estonia. Accessed: September 2017. [Online]. Available: https://e-estonia.com/

[318] Communications Authority of Kenya (CA). (2017) National PKI. Communications Authority of Kenya (CA). Accessed: September 2017. [Online]. Available: http://www.ke-cirt.go.ke/index.php/services/national-pki/

[319] Integrated Government Philippines (iGovPhil). (2017) Philippine National Public Key Infrastructure (PNPKI). Accessed: September 2017. [Online]. Available: http://i.gov.ph/pnpki/

[320] G. Madlmayr, J. Langer, and J. Scharinger, "Managing an NFC ecosystem," in *Mobile Business, 2008. ICMB'08. 7th International Conference on.* IEEE, 2008, pp. 95–101.

[321] Android. (2017) Android Pay. Google. Accessed: September 2017. [Online]. Available: https://www.android.com/intl/en_uk/pay/

[322] SEEK. (2017) SEEK for Android project. maintainted by Giesecke & Devrient GmbH. Accessed: September 2017. [Online]. Available: http://seek-for-android.github.io

[323] H. Heuler. (2014, November) Africa's new thin SIM cards: The line between banks and telcos just got thinner. African Enterprise. Accessed Feb 2017. [Online]. Available: http://www.zdnet.com/article/africas-new-thin-sim-cards-the-line-between-banks-and-telcos-just-got-thinner/

[324] K. Baqer, J. Bezuidenhoudt, R. Anderson, and M. Kuhn, "SMAPs: Short Message Authentication Protocols," in *24th International Workshop on Security Protocols*. Springer LNCS, 2017.

[325] GSMA. (2014, August) Generic Overlay SIM Security Assessment. Accessed: September 2017. [Online]. Available: https://www.gsma.com/publicpolicy/wp-content/uploads/2014/08/GSMA-Security-Group-Overlay_SIM_Security_Assessment_August_18_2014.pdf

[326] R. Metz. (2017, January) Second Life Is Back for a Third Life, This Time in Virtual Reality. MIT Technology Review. Accessed: September 2017. [Online]. Available: https://www.technologyreview.com/s/603422/second-life-is-back-for-a-third-life-this-time-in-virtual-reality/

[327] RFC 4226. HOTP: An HMAC-Based One-Time Password Algorithm, December 2005. http://www.ietf.org/rfc/rfc4226.txt.

[328] OATH. (2017) Initiative for Open Authentication . OATH Authentication, LLC. Accessed: September 2017. [Online]. Available: https://openauthentication.org/

[329] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[330] C. J. F. Cremers, *Scyther: Semantics and verification of security protocols*. Eindhoven University of Technology Eindhoven, Netherlands, 2006.

[331] C. Cremers and S. Mauw, *Operational semantics and verification of security protocols*. Springer Science & Business Media, 2012.

[332] C. J. Cremers, S. Mauw, and E. P. de Vink, "Injective synchronisation: an extension of the authentication hierarchy," *Theoretical Computer Science*, vol. 367, no. 1, pp. 139–161, 2006.

[333] G. Lowe, "A hierarchy of authentication specifications," in *Computer security foundations workshop, 1997. Proceedings., 10th*. IEEE, 1997, pp. 31–43.

[334] C. Cremers, "Scyther User Manual. Department of Computer Science, University of Oxford," 2014.

Appendix A

Supplementary Information

This Appendix gives additional information pertaining to the use cases discussed in this thesis.

A.1 Background

A.1.1 One Time Passwords (OTP)

Characteristics desirable in a One Time Password (OTP) are that it should be easy to compute, but very difficult to identify credentials used to create it. Typical OTP lengths are 8 digits or 6 alphanumeric characters.

OTPs fall into two categories: they use either

1. the **HMAC-Based (HOTP)** algorithm based on the HMAC-SHA-1 which is applied to an increasing counter value. This is converted to shorter, user-friendly values using a Truncate function with K (shared secret) and C (counter value)

   HOTP(K,C) = Truncate(HMAC-SHA-1(K,C))

   Relevant standards are RFC4226 [327] and RFC2289 [272]. Or

2. the **Time-Based (TOTP)** variant of the HOTP algorithm, where a time variable T, replaces the counter C in the HOTP computation. This is standardised in RFC6238 [271].

Hardware Tokens that generates HOTP passcodes when the user requests are known as "event-driven", and these codes remain valid until used by the authenticating application: TOTP tokens generate new codes automatically after a set period of time, which limits the time available to use the OTP. Hardware tokens can be security certified by The Initiative for Open Authentication (OATH) [328].

A.2 EV-2: Supplementary Information

A.2.1 Estonian I-Voting - 2017 Framework

The Estonian I-voting system was updated after the 2015 Elections, and a report detailing the changes was published in June 2017 [19]. The main changes are the inclusion of individual vote-verification features, and verification by third party (auditors) who are now able to check all input/output files throughout the process. New terminology is used in the revised system: "Collector" is the server that lists candidates, checks voters' digital signatures, performs vote verification processes and passes the vote onto the "Processor", where various administrative tasks can be carried out such as annulling repeated i-votes. The Processor mixes the votes and and forwards them to the "Tallier" for counting. The new system is illustrated in Figure A.1.

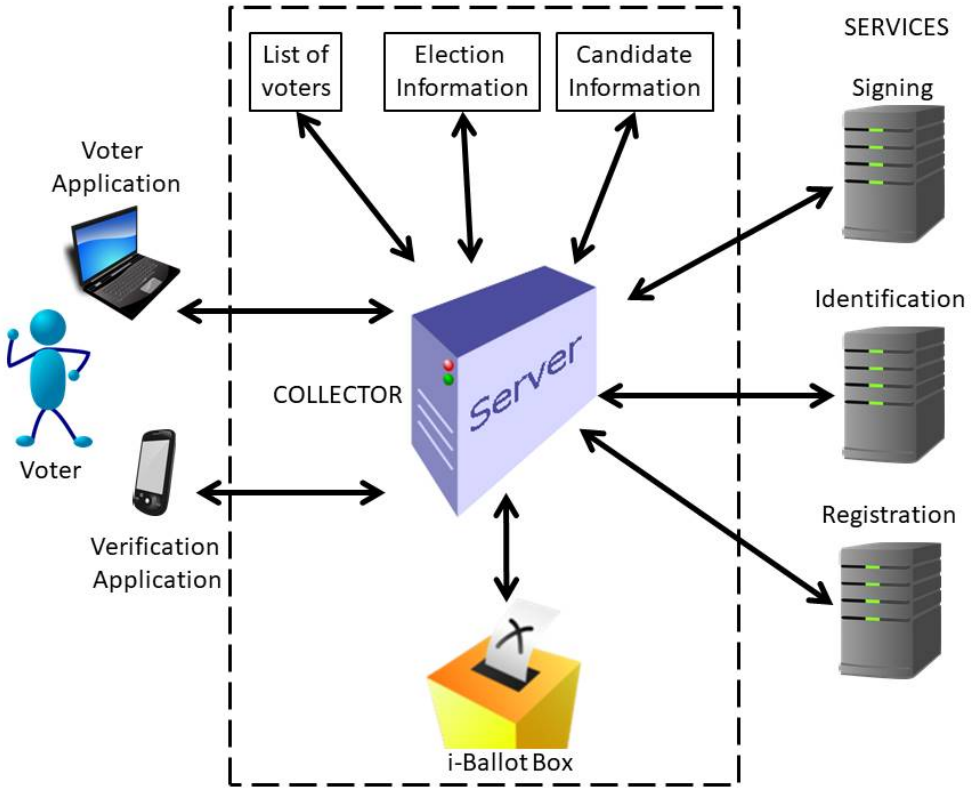

Figure A.1: EV-2: Estonian I-voting (post-2015) - Services and Components (adapted from [19]

Voters can verify their vote using a separate smart device with camera and internet connectivity: this device should not have been used to cast the vote. As seen in Section 4.5.1, the vote is constructed thus: *(candidate choice, random number)* encrypted with the public key of the VCA - i.e. a cryptogram. The cryptogram is signed with the voter's signing key S_V, and the voter's certificate $Cert_V$ is added before sending to the I-voting system server. The server generates a session code, and returns the random number and session code to the voter application to be displayed as a QR code. The verification application on the smart device reads the QR code, extracts the random number and session code, then sends the session code to the I-voting server. The server returns the voter's digitally signed vote to the smart device along with the list of candidates. The verification application then creates cryptograms for all the candidates using the random number, and once it finds one which matches the vote received from the i-voting server then the vote is verified successfully.

A.3 MP-2: Supplementary Information

A.3.1 Bitcoin Transaction Processing

The two Bitcoin payment processing methods relevant to this use case are *Option 1: Multi-Signature Addresses* and *Option 2: Smart Contracts*. The full protocol is shown in Figure A.2 and is now described using notation shown in Table A.1.

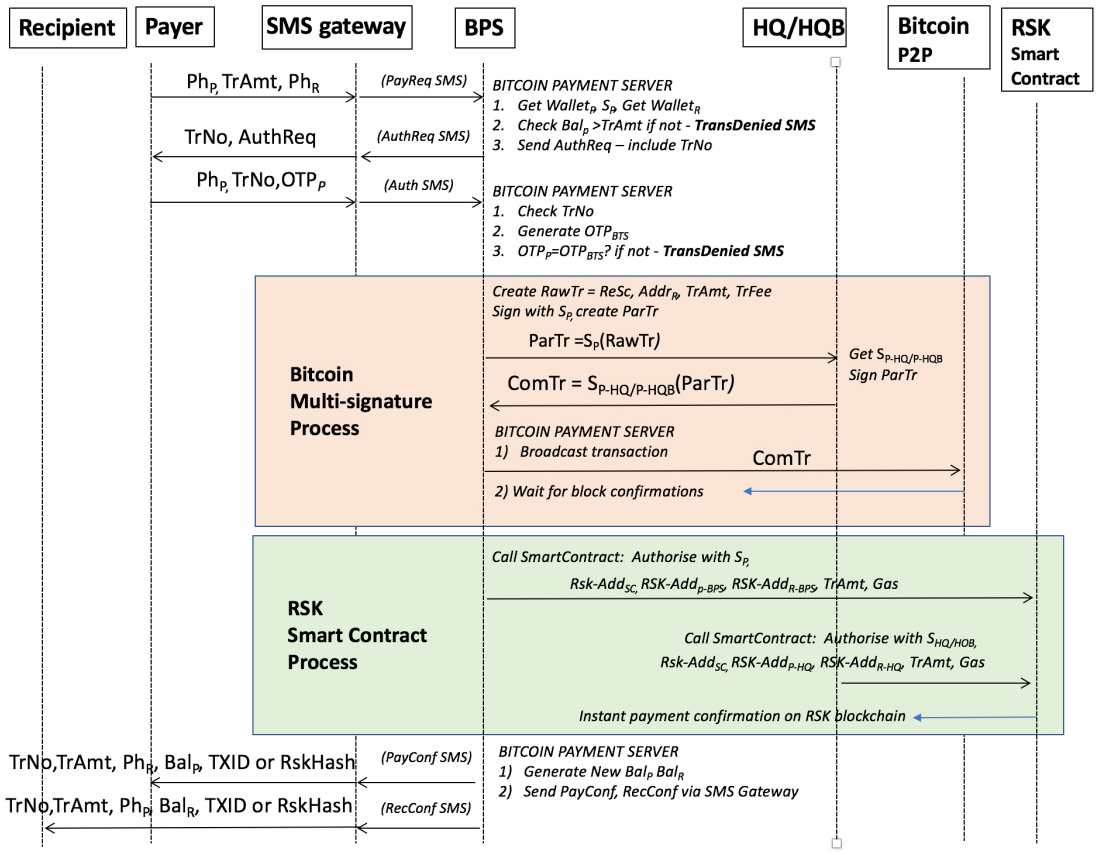

Figure A.2: MP-2: Bitcoin SMS m-Payment - Full Protocol

Option 1: Multi-signature Process

The BPS first generates a Raw Transaction (*RawTr*) which includes the Full Redeem Script (*ReSc*), the new multi-signature address associated for the receiver where the payment is going to, *TrAmt* and *TrFee*. The *RawTr* then needs to be signed by minimum 2 participants in turn to generate a valid Bitcoin transaction. The BPS first signs the *RawTr* using the corresponding Payer private key S_P and forwards the Partial Signed Transaction (*ParTr*) to the HQ for signing.

BPS→HQ: $ParTr = (ReSc, Addr_R, TrAmt, TrFee)Sign_{S_P}$

Table A.1: MP-2: Full Protocol Notation

Notation	Description
$Addr_X$	Bitcoin Multi-signature Address for entity X
BPS	Bitcoin Payment Server(entity)
BAL_X	Bitcoin balance in Account AC_X for entity X
BAL'	Updated Bitcoin balance in Account AC_X for entity X
E_K(Z)	Encryption of data Z with key K
HQ	Head Quarters (entity)
HQB	Head Quarters Backup Location (entity)
LO	Local Office (entity)
OTP_X	One Time Password generated by entity X
P	Payer(entity)
Ph_X	Phone Number of entity X
PK_X/ SK_X	Public/ Secret Key pair of entity X
R	Recipient(entity)
S_X/ V_X	Signing/ Verification key pair of entity X (Bitcoin keys)
TrAmt	Transaction Amount
TrNo	Transaction Number
TXID	Unique Transaction ID of a transaction recorded in the blockchain. Also referred to as the Transaction Hash (TrHash)
TrHash	Transaction Hash
X→Y:	Message sent from entity X to entity Y
(Z)$Sign_K$	Signature on data Z with signature key K
TrFee	Transaction Fee paid to the Bitcoin miner
RawTr	Raw Transaction created for signing
ParTr	Partial Signed Transaction created after signing *RawTr*
ComTr	Complete Signed Transaction created after signing *ParTr*
ReSc	Full Redeem Script used for the Bitcoin multi-signature address
RSKHash	Rootstock Transaction Hash
RSK-Add_{SC}	RSK Smart Contract Address, unique for the contract and never changes
RSK-Add_{X-Y}	RSK public key (RSK address) of entity X kept securely with entity Y
$S_{RSK-X-Y}$	RSK private key of entity X kept securely with entity Y
Gas	Transaction fee paid to execute instructions on the smart contract

To authorise the payment request, HQ first verifies the *ParTr* to check the payment amount and number of signatures needed. Once satisfied, HQ signs this using its private payer Bitcoin key S_{P-HQ} to generate the Complete Signed Transaction *ComTr* and sends this back to the BPS.

HQ→BPS: ComTr $=(ParTr)Sign_{S_{P-HQ}}$

The BPS then broadcasts the *ComTr* to the Bitcoin peer-to-peer network. Once broadcast, a unique transaction-id (TXID) or the recipient's Bitcoin address can be used to trace the transaction on the blockchain. The Bitcoin miner who first publishes the valid block in the blockchain that also includes our Bitcoin transaction is paid the *TrFee* for the payment. This is the first confirmation for the transaction. The BPS then waits for the transaction to be confirmed in the agreed number of blocks before generating the SMSs.

Option 2: Smart Contract Process

The BPS calls the Smart Contract and authorises the *TrAmt* and the fee for executing the transaction also called *Gas* is paid by using the $S_{RSK-P-BPS}$.
$BPS \rightarrow RSK: = RSK\text{-}Add_{SC}, RSK\text{-}Add_{P}, RSK\text{-}Add_{R}, TrAmt, Gas$

Once the message gets broadcast in the RSK network, the HQ or the HQB calls the smart contract which act as the second set of instructions needed by the smart contract to execute the transaction. HQ/HQB uses the $S_{RSK-P-HQ/HQB}$ to authorise the paid amount *TrAmt* and the transaction fee *Gas*.
$HQ/HQB \rightarrow RSK: = RSK\text{-}Add_{SC}, RSK\text{-}Add_{P-HQ}, RSK\text{-}Add_{R}, TrAmt, Gas$

When instructions are received from both BPS and HQ/HQB, the Smart Contract executes a transaction to transfer the value *TrAmt* to the recipient. The unique transaction details are recorded instantly on the RSK blockchain in the format of a hash (RSKHash). The BPS does not need to wait for a transaction confirmation as there is instant confirmation when using the RSK platform.

Appendix B

Scyther Scripts

This Appendix describes the formal security analysis tool Scyther, and lists Scyther scripts and verification results for various protocols presented in this thesis.

B.1 The Scyther Formal Security Analysis Tool

Scyther performs an automatic formal analysis of security protocols under the perfect cryptography assumption (the Dolev-Yao model, where it is assumed that an adversary can learn nothing from an encrypted message unless they are in possession of the relevant decryption key [329]), for an unbounded number of instances [30, 330]. The Scyther tool can be used to find problems that arise from the way a protocol has been constructed.

Scyther allows users to verify protocols based on the security properties defined in an input file (using the *Verify protocol* option). It is possible to vary verification parameters such as maximum number of runs, matching type and types of attacks to search for. A pop-up window gives the results of the verification, with clickable links to graphical representations of any attacks found. By convention, Scyther protocol description files (scripts) have the extension .spdl (Security Protocol Description Language).

B.1.1 Scyther Roles, Events and Claims

Scyther models security properties via roles. Each entity involved in a protocol is considered as one role, and events and claims are made for each role. Events are messages that are exchanged between entities: each "send" event within a role must have a corresponding "receive" event specified in the receiving role. There is a facility to check if the roles can complete the protocol i.e. they are "reachable" (by using the *Characterise role* option). Claims are the security properties to be verified, based on the entity's local view of the state of the system.

A Scyther script starts with function declarations, and then events and claims are set out for each role.

Authentication properties are verified through agreement, aliveness and synchronisation. Claims that can be included in scripts are as follows:

- Secrecy (*Secret*): this verifies confidentiality of secret keys or data.
- Weak agreement (*WeakAgree*): this verifies the data exchanged between entities.
- Aliveness (*Alive*): verifies the authentication of communication partners
- Non-injective synchronisation (*Nisynch*): verifies that entities know who they are communicating with, and agree on the content and order of messages. This is a stronger authentication requirement than aliveness.
- Non-injective agreement (*Niagree*): verifies that communicating entities agree on the content of variables.

Definitions of these properties can be found in the works of Cremers et al. [331, 332] and Lowe [333].

B.1.2 Verification Results

A description of the results that can be obtained after verifying claims can be seen in the Scyther User Manual [334]. Results may include the following:

- verified as "OK" in the "Status" column and "No attacks within bounds" in the "Comments". This means that no attack was found within the bounded statespace but there may be attacks that can occur outside the bounded statespace.
- verified as "OK" in the "Status" with "Verified" and "No attacks" in the "Comments", this means that no attack was found within the bounded or unbounded statespace; the security property has been successfully verified..
- "Status" can show "falsified", which means at least one attack on the protocol is possible

.

B.2 EV-1/EV-2: SCWS Remote e-Voting

This script is for the SCWS generic e-Voting protocol described in Chapter 4.1, Section 4.3. There are two roles specified in the script, RAS and SIM, and claims are made about the secrecy of credentials, along with Non-injective agreement (Niagree), Non-injective synchronisation (Nisynch) and Aliveness (Alive). The Scyther verification result (**No attacks within bounds**) is shown in Figure B.1.

B.2.1 EV1/EV2: SCWS e-Voting Generic Model Protocol

```
usertype PublicKey;
usertype PrivateKey;
usertype Vote;
usertype String;
usertype Application;


secret https: Function;
```

```
protocol Generic-eVoting(SIM, RAS) {

     role RAS {

        const PUBV: PublicKey;
        const PUBVA: PublicKey;
        const PKV: PrivateKey;
        const VoterID: String;
        const passWord: String;
        const VoteApp: Application;

        var vote:Vote;

        send_1(RAS,SIM, https(PUBV, PUBVA, PKV));
not match (RAS,SIM);

        send_2(RAS,SIM,  https({VoterID, passWord, VoteApp}PUBVA) );
        recv_3(SIM,RAS,  https({vote}PUBVA));
claim_RAS0(RAS,Running,SIM,vote);


        claim_RAS1(RAS, Secret, PKV);
        claim_RAS2(RAS, Secret, VoterID);
        claim_RAS3(RAS, Secret, passWord);
        claim_RAS4(RAS, Secret, VoteApp);
        claim_RAS5(RAS, Secret, vote);

        claim_RAS6(RAS, Niagree);
        claim_RAS7(RAS, Nisynch);
        claim_RAS8(RAS, Alive);
claim_RAS9(RAS,Weakagree);
    }

     role SIM {

        const vote:Vote;
```

```
        var PUBV: PublicKey;
        var PUBVA: PublicKey;
        var PKV: PrivateKey;
        var VoterID : String;
        var passWord: String;
        var VoteApp: Application;

        recv_1(RAS,SIM, https(PUBV, PUBVA, PKV));
        recv_2(RAS,SIM, https({VoterID, passWord, VoteApp}PUBVA) );
        send_3(SIM,RAS, https({vote}PUBVA));
not match (SIM,RAS);
claim_SIM0(SIM,Commit,RAS,vote);

        claim_SIM1(SIM, Secret, PKV);
        claim_SIM2(SIM, Secret, VoterID);
        claim_SIM3(SIM, Secret, passWord);
        claim_SIM4(SIM, Secret, VoteApp);
        claim_SIM5(SIM, Secret, vote);

        claim_SIM6(SIM, Niagree);
        claim_SIM7(SIM, Nisynch);
        claim_SIM8(SIM, Alive);
claim_SIM9(SIM,Weakagree);

    }

}
```

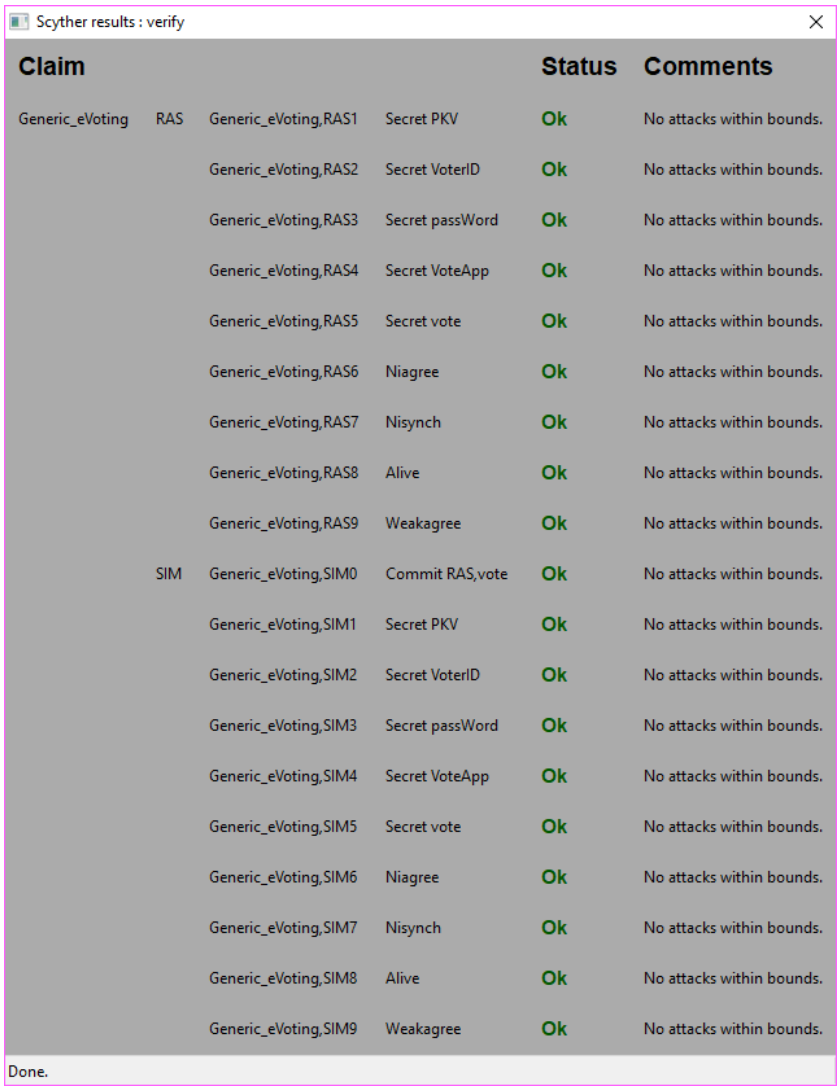

Figure B.1: EV-1/EV-2: SCWS Generic e-Voting Protocol - Scyther Verification

B.3 MP-1: SCWS Branchless Banking

These scripts are for the SCWS Branchless Banking protocols (Withdrawal, Deposit and Transfer) described in Chapter 5, Sections 5.3.4, 5.3.5 and 5.3.6. The three roles specified for Withdrawal and Deposit are Agent (A), Bank (B) and Customer (C): Transfer has Bank (B) Customer (C) and Recipient (R). Claims are made about the secrecy of credentials, along with Non-injective agreement (Niagree), Non-injective synchronisation (Nisynch) and Aliveness (Alive). The Scyther verification results are shown in Figure B.2, Figure B.3 and Figure B.4 respectively. The results are either (**No attacks within bounds**) or **Verified No Attacks**.

B.3.1 MP-1: SCWS-Banking Withdrawal Protocol

```
/*
 * Withdrawal protocol
 */


secret https: Function;

// The protocol description

protocol BBwithdrawal(A,B,C)

{

role C
{
fresh nc: Nonce;
fresh TrCount: Nonce;
var nc1: Nonce;
var TrNo: Data;
var BalC: Data;


const Trans: Data; // Tr, TrAmt

send_1 (C,B, https({C,B,{Trans,A,nc,TrCount}pk(B)}sk(C)) );
not match (A,C);
not match (A,B);
not match (B,C);


recv_5(B,C, https({{Trans, TrNo, A,nc1,BalC}pk(C)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);
```

```
claim(C,Secret,nc);
claim(C,Secret,nc1);

claim(C,Alive);
claim(C,Weakagree);
claim(C,Niagree);
claim(C,Nisynch);

}

role B
{
fresh na1: Nonce;
fresh na2: Nonce;
fresh nb: Nonce;
fresh nb: Nonce;
fresh nb2: Nonce;
fresh nc1: Nonce;

var Trans: Data;
var TrCount: Nonce;
var nc: Nonce;
var nb1: Nonce;
var na: Nonce;
var ChC: Data;

const TrNo: Data;
const NameC: Data;
const BalC: Data;
const BalA: Data;
const BalA1: Data;


recv_1(C,B,https({C,B,{Trans,A,nc,TrCount}pk(B)}sk(C)) );
not match (A,C);
not match (A,B);
not match (B,C);
```

```
send_2 (B,A, https({B,A,{C, TrNo,Trans,NameC,BalA,nb}pk(A)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);


recv_3 (A,B, https({A,B,{TrNo,Trans,ChC,nb1,na}pk(B)}sk(A)) );
not match (A,C);
not match (A,B);
not match (B,C);


    send_4(B,A, https({B,A,{Trans,C,nb2,na1,BalA1}pk(A)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);

send_5(B,C, https({{Trans, TrNo, A,nc1,BalC}pk(C)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);


claim(B,Secret,nc);
claim(B,Secret,nc1);

claim(B,Secret,na);
claim(B,Secret,na1);
claim(B,Secret,nb);
claim(B,Secret,nb1);
claim(B,Secret,nb2);

claim(B,Alive);
claim(B,Weakagree);
claim(B,Niagree);
```

```
claim(B,Nisynch);


}

role A
{
fresh na: Nonce;
fresh nb1: Nonce;
fresh ChC: Data;

var nb: Nonce;
var nb2: Nonce;
var na1: Nonce;
var Trans: Data;
var TrNo: Data;
var NameC: Data;
var BalA: Data;
var BalA1: Data;

recv_2(B,A, https({B,A,{C, TrNo,Trans,NameC,BalA,nb}pk(A)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);

send_3 (A,B, https({A,B,{TrNo,Trans,ChC,nb1,na}pk(B)}sk(A)) );
not match (A,C);
not match (A,B);
not match (B,C);


recv_4(B,A, https({B,A,{Trans,C,nb2,na1,BalA1}pk(A)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);
```

```
claim(A,Secret,na);
claim(A,Secret,na1);
claim(A,Secret,nb);
claim(A,Secret,nb1);
claim(A,Secret,nb2);

claim(A,Alive);
claim(A,Weakagree);
claim(A,Niagree);
claim(A,Nisynch);


}

}
```

BBwithdrawal	C	BBwithdrawal,C7	Secret nc	Ok	Verified	No attacks.
		BBwithdrawal,C8	Secret nc1	Ok	Verified	No attacks.
		BBwithdrawal,C9	Alive	Ok	Verified	No attacks.
		BBwithdrawal,C10	Weakagree	Ok	Verified	No attacks.
		BBwithdrawal,C11	Niagree	Ok	Verified	No attacks.
		BBwithdrawal,C12	Nisynch	Ok	Verified	No attacks.
	B	BBwithdrawal,B16	Secret nc	Ok	Verified	No attacks.
		BBwithdrawal,B17	Secret nc1	Ok	Verified	No attacks.
		BBwithdrawal,B18	Secret na	Ok	Verified	No attacks.
		BBwithdrawal,B19	Secret na1	Ok	Verified	No attacks.
		BBwithdrawal,B20	Secret nb	Ok	Verified	No attacks.
		BBwithdrawal,B21	Secret nb1	Ok	Verified	No attacks.
		BBwithdrawal,B22	Secret nb2	Ok	Verified	No attacks.
		BBwithdrawal,B23	Alive	Ok	Verified	No attacks.
		BBwithdrawal,B24	Weakagree	Ok	Verified	No attacks.
		BBwithdrawal,B25	Niagree	Ok	Verified	No attacks.
		BBwithdrawal,B26	Nisynch	Ok	Verified	No attacks.
	A	BBwithdrawal,A10	Secret na	Ok		No attacks within bounds.
		BBwithdrawal,A11	Secret na1	Ok		No attacks within bounds.
		BBwithdrawal,A12	Secret nb	Ok		No attacks within bounds.
		BBwithdrawal,A13	Secret nb1	Ok		No attacks within bounds.
		BBwithdrawal,A14	Secret nb2	Ok		No attacks within bounds.
		BBwithdrawal,A15	Alive	Ok		No attacks within bounds.
		BBwithdrawal,A16	Weakagree	Ok		No attacks within bounds.
		BBwithdrawal,A17	Niagree	Ok		No attacks within bounds.
		BBwithdrawal,A18	Nisynch	Ok		No attacks within bounds.

Done.

Figure B.2: MP-1: Withdrawal Protocol - Scyther Verification

B.3.2 MP-1: SCWS-Banking Deposit Protocol

```
/*
 *Deposit protocol
 */

secret https: Function;

// The protocol description

protocol BBdeposit(A,B,C)
{
role A
{

fresh na: Nonce;
fresh TrCount: Nonce;
var na1: Nonce;
var TrNo: Data;
var BalA: Data;
const Trans: Data; // Tr, TrAmt

send_1 (A,B, https({A,B,{Trans,C,na,TrCount}pk(B)}sk(A)) );
not match (A,C);
not match (A,B);
not match (B,C);

recv_5(B,A, https({B,A,{TrNo,Trans, C,na1,BalA}pk(A)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);

claim(A,Secret,na);
claim(A,Secret,na1);

claim(A,Alive);
claim(A,Weakagree);
claim(A,Niagree);
```

```
claim(A,Nisynch);

}

role B
{

fresh na1: Nonce;
fresh nb: Nonce;
fresh nb2: Nonce;
fresh nc1: Nonce;

var Trans: Data;
var TrCount: Nonce;
var nc: Nonce;
var nb1: Nonce;
var na: Nonce;
var ChA: Data;
const TrNo: Data;
const BalA: Data;
const BalC: Data;
const BalC1: Data;

recv_1 (A,B, https({A,B,{Trans,C,na,TrCount}pk(B)}sk(A)) );
not match (A,C);
not match (A,B);
not match (B,C);

send_2 (B,C, https({B,C,{TrNo,Trans, A,BalC,nb}pk(C)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);

recv_3 (C,B, https({C,B,{TrNo,Trans,ChA,nb1,nc}pk(B)}sk(C)) );
not match (A,C);
not match (A,B);
not match (B,C);
```

```
send_4(B,C, https({B,C,{TrNo,Trans,A,nb2,nc1,BalC1}pk(C)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);

send_5(B,A, https({B,A,{TrNo,Trans, C,na1,BalA}pk(A)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);

claim(B,Secret,na);
claim(B,Secret,na1);
claim(B,Secret,nb);
claim(B,Secret,nb1);
claim(B,Secret,nb2);
claim(B,Secret,nc);
claim(B,Secret,nc1);

claim(B,Alive);
claim(B,Weakagree);
claim(B,Niagree);
claim(B,Nisynch);


}

role C
{
fresh nc: Nonce;
fresh nb1: Nonce;
fresh ChA: Data;

var nb: Nonce;
var nb2: Nonce;
var nc1: Nonce;
var Trans: Data;
```

```
var TrNo: Data;
var BalC: Data;
var BalC1: Data;

recv_2(B,C, https({B,C,{ TrNo,Trans, A,BalC,nb}pk(C)}sk(B)));
not match (A,C);
not match (A,B);
not match (B,C);

send_3 (C,B, https({C,B,{TrNo,Trans,ChA,nb1,nc}pk(B)}sk(C)) );
not match (A,C);
not match (A,B);
not match (B,C);

recv_4(B,C, https({B,C,{TrNo, Trans,A,nb2,nc1,BalC1}pk(C)}sk(B)) );
not match (A,C);
not match (A,B);
not match (B,C);

claim(C,Secret,nc);
claim(C,Secret,nc1);
claim(C,Secret,nb);
claim(C,Secret,nb1);
claim(C,Secret,nb2);

claim(C,Alive);
claim(C,Weakagree);
claim(C,Niagree);
claim(C,Nisynch);

}

}
```

BBdeposit	A	BBdeposit,A7	Secret na	Ok	Verified	No attacks.
		BBdeposit,A8	Secret na1	Ok	Verified	No attacks.
		BBdeposit,A9	Alive	Ok	Verified	No attacks.
		BBdeposit,A10	Weakagree	Ok	Verified	No attacks.
		BBdeposit,A11	Niagree	Ok	Verified	No attacks.
		BBdeposit,A12	Nisynch	Ok	Verified	No attacks.
	B	BBdeposit,B16	Secret na	Ok	Verified	No attacks.
		BBdeposit,B17	Secret na1	Ok	Verified	No attacks.
		BBdeposit,B18	Secret nb	Ok	Verified	No attacks.
		BBdeposit,B19	Secret nb1	Ok	Verified	No attacks.
		BBdeposit,B20	Secret nb2	Ok	Verified	No attacks.
		BBdeposit,B21	Secret nc	Ok	Verified	No attacks.
		BBdeposit,B22	Secret nc1	Ok	Verified	No attacks.
		BBdeposit,B23	Alive	Ok	Verified	No attacks.
		BBdeposit,B24	Weakagree	Ok	Verified	No attacks.
		BBdeposit,B25	Niagree	Ok	Verified	No attacks.
		BBdeposit,B26	Nisynch	Ok	Verified	No attacks.
	C	BBdeposit,C10	Secret nc	Ok		No attacks within bounds.
		BBdeposit,C11	Secret nc1	Ok		No attacks within bounds.
		BBdeposit,C12	Secret nb	Ok		No attacks within bounds.
		BBdeposit,C13	Secret nb1	Ok		No attacks within bounds.
		BBdeposit,C14	Secret nb2	Ok		No attacks within bounds.
		BBdeposit,C15	Alive	Ok	Verified	No attacks.
		BBdeposit,C16	Weakagree	Ok	Verified	No attacks.
		BBdeposit,C17	Niagree	Ok		No attacks within bounds.
		BBdeposit,C18	Nisynch	Ok		No attacks within bounds.

Done.

Figure B.3: MP-1: Deposit Protocol - Scyther Verification

B.3.3 MP-1: SCWS-Banking Transfer Protocol

```
/*
 * BB Transfer protocol
 */

secret https: Function;

// The protocol description

protocol BBTransfer(C,B,R)

{

role C
{
fresh nc: Nonce;
fresh TrCount: Nonce;
var nc1: Nonce;
var TrNo: Data;
var BalC: Data;

const Trans: Data; // Tr, TrAmt

send_1 (C,B, https({C,B,{Trans,R,nc,TrCount}pk(B)}sk(C)) );
not match (R,C);
not match (R,B);
not match (B,C);

recv_3(B,C, https({{Trans, TrNo, R,nc1,BalC}pk(C)}sk(B)) );
not match (R,C);
not match (R,B);
not match (B,C);

claim(C,Secret,nc);
claim(C,Secret,nc1);

claim(C,Alive);
```

```
claim(C,Weakagree);
claim(C,Commit,B,nc);
claim(C,Niagree);
claim(C,Nisynch);

}

role B
{
fresh nc1: Nonce;

var Trans: Data;
var TrCount: Nonce;
var nc: Nonce;

const TrNo: Data;
const NameC: Data;
const BalC: Data;
const BalR: Data;

recv_1 (C,B, https({C,B,{Trans,R,nc,TrCount}pk(B)}sk(C)) );

not match (R,C);
not match (R,B);
not match (B,C);

send_2 (B,R, https({B,R,{TrNo,Trans,NameC,BalR}pk(R)}sk(B)) );
not match (R,C);
not match (R,B);
not match (B,C);

send_3(B,C, https({{Trans, TrNo, R,nc1,BalC}pk(C)}sk(B)) );
not match (R,C);
not match (R,B);
not match (B,C);

claim(B,Secret,nc);
```

```
claim(B,Secret,nc1);

claim(B,Alive);
claim(B,Weakagree);
claim(B,Niagree);
claim(B,Nisynch);

}

role R
{

var Trans: Data;
var TrNo: Data;
var NameC: Data;
var BalR: Data;

recv_2 (B,R, https({B,R,{TrNo,Trans,NameC,BalR}pk(R)}sk(B)) );
not match (R,C);
not match (R,B);
not match (B,C);


claim(R,Alive);
claim(R,Weakagree);
claim(R,Niagree);
claim(R,Nisynch);

}

}
```

Figure B.4: MP-1: Transfer Protocol - Scyther Verification

B.4 Auth-1: SCWS Single Sign-On

This script is for the SCWS Single Sign On protocol described in Chapter 6, Section 6.3. There are three roles specified in the script, USER, SIM and MOD, and claims are made about the secrecy of credentials, along with Non-injective agreement (Niagree), Non-injective synchronisation (Nisynch) and Aliveness (Alive). The Scyther verification result (**No attacks within bounds**) is shown in Figure B.5.

B.4.1 Auth-1: SCWS Single Sign-On Protocol

```
/*
 *SSO protocol
```

```
 */

secret https: Function;

// The protocol description

protocol SSO(USER,SIM,MOD)

{

role USER
{
var Info: Data;

const Password: Data;
const IDUSER: Data;

send_1 (USER,SIM, https(IDUSER,Password) );
not match (USER,SIM);

recv_4(SIM,USER, https(Info));
not match (USER,SIM);

claim(USER,Commit,SIM,Password);

claim(USER,Alive);
claim(USER,Weakagree);
claim(USER,Niagree);
claim(USER,Nisynch);

}

role SIM
{
fresh T: Data;

var Password: Data;
```

```
var Info: Data;
var IDUSER: Data;

const IDSIM: Data;
const L:Data;

recv_1 (USER,SIM, https(IDUSER,Password) );
not match (USER,SIM);
claim(SIM,Running,USER,Password);

send_2 (SIM,MOD, https(SIM,MOD,{L,T,IDSIM}sk(SIM) ));
not match (SIM,MOD);

recv_3 (MOD,SIM, https(Info) );
not match (SIM,MOD);

send_4 (SIM,USER, https(Info) );
not match (SIM,USER);
claim(SIM,Commit,MOD,L,T,IDSIM);

claim(SIM,Secret, L);
claim(SIM,Secret, T);
claim(SIM,Secret,IDSIM);
claim(SIM,Secret, Info);

claim(SIM,Alive);
claim(SIM,Weakagree);
claim(SIM,Niagree);
claim(SIM,Nisynch);
}

role MOD
{
const Info: Data;

var T: Data;
var IDSIM: Data;
```

```
var L:Data;

recv_2 (SIM,MOD, https(SIM,MOD,{L,T,IDSIM}sk(SIM)) );
not match (SIM,MOD);

send_3 (MOD,SIM, https(Info) );
not match (SIM,MOD);

claim(MOD,Running,SIM,L,T,IDSIM);

claim(MOD,Secret, L);
claim(MOD,Secret, T);
claim(MOD,Secret,IDSIM);
claim(MOD,Secret, Info);

claim(MOD,Alive);
claim(MOD,Weakagree);
claim(MOD,Niagree);
claim(MOD,Nisynch);

}

}
```

Figure B.5: Auth-1: Single Sign-On Protocol - Scyther Verification