COMPARING AND CONTRASTING DIFFERENT MOBILE PHONE TECHNOLOGIES WHEN IMPLEMENTING OUT-OF-BAND AUTHENTICATION TO A WEB PORTAL USING SOCIAL SECURITY NUMBERS TO IDENTIFY USERS.

Ashley Wagstaffe

Submitted in partial fulfilment of the requirements of Napier University for the Degree of BEng (Hons) Computer Networks and Distributed Systems

School of Computing
December 2008
Authorship Declaration

I, Ashley Wagstaffe, confirm that this Thesis and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this Thesis is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work of this dissertation following the School’s ethical guidelines

Signed:

Date:

Matriculation no:

PLEASE NOTE that in signing this page you are aware of the consequences of doing this fraudulently as explained at

http://www.napier.ac.uk/ed/plagiarism/homepage.htm
Data Protection Declaration

Under the 1998 Data Protection Act, The University cannot disclose your grade to an unauthorised person. However, other students benefit from studying theses that have their grades attached.

The University may make this Thesis with indicative grade, available to others.

The University may make this Thesis available to others, but the grade may not be disclosed.

The University may not make this Thesis available to others.
Contents

1 INTRODUCTION 10
  1.1 BACKGROUND AND MOTIVATION 10
  1.2 KEY AIM AND OBJECTIVES 12
  1.3 APPROACH AND FUTURE GOALS 13
  1.4 THESIS OUTLINE 13

2 LITERATURE REVIEW 14
  2.1 IDENTITY MANAGEMENT 14
    2.1.1 History of identity management 14
    2.1.2 Identity Theft 15
  2.2 ONLINE AUTHENTICATION PROCESSES 17
    2.2.1 Authentication systems 18
  2.3 MOBILE DEVICE TECHNOLOGIES 22
    2.3.1 Mobile cellular networks 22
    2.3.2 Internet Connectivity 23
    2.3.3 Windows Mobile 6.1 24
    2.3.4 Java Micro Edition (JavaMe) 25
  2.4 CONCLUSIONS 27
    2.4.1 The Identity Theft problem 27
    2.4.2 The March toward Managed Cards 27
    2.4.3 Mobile Technologies 27

3 REQUIREMENTS ANALYSIS 28
  3.1 INTRODUCTION 28
  3.2 IDENTITY METASYSTEM COMPONENTS 29
  3.3 IDENTITY PROVIDER COMPONENTS 30
    3.3.1 Secure Token Server (STS) 30
    3.3.2 Attributes Store (Registered Users Store) 30
  3.4 SIMPLE MESSAGE SERVICE CENTRE (SMSC) 31
  3.5 COMMUNICATION PROTOCOLS 32
    3.5.1 Web Service Protocols 82
    3.5.2 WS-Policy version 1.2 82
    3.5.3 WS-Metadata Exchange 82
    3.5.4 WS-Security (WSS) 83
    3.5.5 WS-Trust 83
    3.5.6 Custom Encryption for OTA communication 83
  3.6 .NET 3.5x WINDOWS COMMUNICATION FOUNDATION (WCF) 32
    3.6.1 Development Tools 32
    3.6.2 Application Programmers Interface (API) 33
  3.7 JAVA ME 34
    3.7.1 Development Tools 34
    3.7.2 API 35
  3.8 CONCLUSION 36

4 DESIGN 37
  4.1 INTRODUCTION 37
  4.2 HIGH-LEVEL ABSTRACTION 37
  4.3 PERSONA / STORYBOARD 38
  4.4 DESIGN CONSIDERATIONS 39
Figures

Figure 1: RSA SecurID® USB Dongle (RSA SecurID, 2008)  
Figure 2: Managed Cards  
Figure 3: Wireless Web technology landscape (Gavalas & Economou, 2007)  
Figure 4: .Net Compact Framework (Microsoft, 2008)  
Figure 5: JavaMe stack (B'Far, 2005)  
Figure 6: Out-of-band self-authentication as part of Identity Provision  
Figure 7: Out-of-band loop in focus  
Figure 8: Identity Provider Abstraction  
Figure 9: Hardware Infrastructure model  
Figure 10: Authorisation Service Activity Diagram  
Figure 11: Metasystem Sequence Diagram  
Figure 12: Intemediaiy Authorisation Service  
Figure 13: Authorisation Service Class Diagram  
Figure 14: Restful Attempt Retrieval Service  
Figure 15: Mobile Client  
Figure 16: WAP Push topology  
Figure 17: WAP 1.0 Protocol Stack (Orange Partner)  
Figure 18: i-Mode Protocol Stack  
Figure 19: WAP 2.0 Protocol Stack (Orange Partner)  
Figure 20: Windows Mobile 6 SDK Documentation (Microsoft, 2008)

Tables

Table 1: Windows Mobile Symmetric Encryption (Microsoft, 2008)  
Table 2: Implemented Servers  
Table 3: RAM Allocation between implemented servers
Code Snippets

Code Snippet 1: Setting the certificates in the STS
Code Snippet 2: Setting the STS certificate in the Web.Config file
Code Snippet 3: Encrypting an Attempt object using Reflection
Code Snippet 4: Activating the application with an SMS message
Code Snippet 5: Mobile Fetching encrypting attempt data and deserialising
Acknowledgements

I would like to thank Professor William J. Buchanan (Bill) and for giving his time, guidance and supervision, Dr Ahmed Al-Dubai for agreeing to be my second marker and both for agreeing to do so at short-notice. I would also like to thank Professor Ben Paechter for supervising me during the first half of this project and wish him well in his new post. Additional thanks goes to Napier University and the Microsoft Academic Alliance Program which greatly assisted with the implementation of this project. I would further like to thank Kim Cameron and John Howie from Microsoft who have provided valuable input into this Thesis and to Barry Dorrans who gave me initial guidance during his break between presentations at Developer Day Scotland 2008. I would like to also thank Alison Fleming who supported me as a true friend during this thesis and to my Father for giving his time to the unenviable task of proof reading. Final thanks are directed at a group of Ramblers on the Island of Mull who helped in the user evaluation of the implemented prototype.
Abstract

With increasing numbers of broadband connections (Office for National Statistics, 2008) and consumers conducting ever more complex transactions on those connections (Nicholas, Kershaw, & Walker, 2006/2007), it is imperative that users and services have accountability through proof of identity (Summers, 1997). Yet some proponents argue that given the openness of the internet it may be almost impossible to absolutely prove the identity of a remote person or service (Price, 2006).

Kim Cameron in his argument for Federated Identity states that “A system that does not put users in control will – immediately or over time – be rejected” (2005) which is also a view echoed by Dean (Identity Management – back to the user, 2006). The aim of the thesis is to argue for a self-authentication factor that is integrated into a Federated Identity infrastructure using an out-of-band loop to a mobile device; this argument is then supported with an implemented proof-of-concept prototype. The prototype and its concept are evaluated in a small usability study and an encryption performance experiment on a mobile device.

The results of the usability study show that users feel more comfortable with self-authentication using something physical that they hold and respond to than with a third party verifying information on their behalf. The results also show the encryption needed for end-to-end confidentiality and integrity during the out-of-band communication will affect battery life to a degree.

The thesis concludes that there is a sound base for self-authentication from a user perspective and that further user and infrastructure studies will need to be conducted on self-authentication before it is realised in the marketplace. It also found that implementing the prototype was more straightforward for the .Net Compact Framework on the Windows Mobile device than it was using the JavaMe platform.
1 Introduction

This aim of this introduction is to present the reader with motives behind this project. The project will be justified by giving a brief background into issues that currently surround online transaction security and that face the stakeholders involved in online transactions. The Deliverables will be presented in this chapter as a Key aim and a list of objectives. The final part of the introduction provides the reader with a summary of the thesis layout and a list of assumptions which include the scope of this Thesis and its related Project.

1.1 Background and Motivation

In the eyes of the developed nations the personal identity of a human being is inexorably linked to the registration of their birth, the name that they are registered with and possibly their nationality and knowledge of their parents. Article 7 clause 1 of the UN Convention on the Rights of the Child states that:

“The child shall be registered immediately after birth and shall have the right from birth to a name, the right to acquire a nationality and, as far as possible, the right to know and be cared for by his or her parents.”

(UN, 1989)

It is worth noting that the UN Human Rights Committee does not yet press these obligations on all countries because it is considered that an identity registration system can be ill afforded by a nation undergoing economic development (Szreter, 2007).

René Descartes, widely regarded as the father of modern philosophy (Metaphysics Research Lab, CSLI, Stanford University, 2005), famously said in 1637 “Cogito, ergo sum”. Brinkmann uses Descartes’ theory in his article (2005) suggesting that the act of a human being’s awareness of thinking points to the identity of that human. The concept of self-narrative in understanding self-identity is explored in Kose’s article (2002) where he uses the existentialist works of Samuel Beckett to illustrate that:
“Identity is grounded and shaped by the canonical story that has so far unfolded, and whose continuation is anticipated”

The identification of self could be said to be as strongly linked to self-identity and existentialism (Currall, Moss, & Stuart, 2008) as it is to an object oriented view of the world and that a human being is a singleton.

If Samuel Beckett and other philosophers are right about self identity and that it bound to this idea of a stream of consciousness that has a past, a present and an imagined future then is it a great leap of faith to say that an authentication process to a remote service should be thought of as a stream of events that has an initiation a middle and a perceived result? Who better than ourselves to determine what we are doing at the current time because we know where we have been and where we perceive we might be? Which could be simply put as either “That’s me that’s doing that” or “Hey! I’m not doing that”.

E-commerce sales by non-financial UK businesses were estimated to have risen 29.1% between 2005 and 2006 (Nicholas, Kershaw, & Walker, 2006 /2007). The ONS also reported that 65% of households in Great Britain had internet access with the majority being broadband connection (Office for National Statistics, 2008). These statistics indicate a rapidly growing uptake by consumers of e-commerce and an implicit growing reliance by the consumer on the authentication strategies used by e-commerce sites.

In order for a successful secure online transaction to take place, proof of identity by the authenticating party to the distributed system is fundamental. This proof of identity fulfils one of the key security principles of accountability (Summers, 1997)

Identity theft is considered to be a growing criminal activity with UK credit agencies like Experian reporting 66% increase in Identity Fraud (Savvas, 2008) and High Street banks are becoming “increasingly reluctant” to compensate ID Fraud victims (Hussain, 2006).
Given the issues outlined above, security of online transactions has become one of the prime concerns of the financial industry today, with the Financial Services Authority (FSA) collaborating with Universities like Napier to reduce the risks of e-crime (Buchanan, Risk Analysis in e-Crime). Moreover there is growing anecdotal evidence that the consumer confidence in the security of online transactions might be waning with more than 255,000 complaints of identity theft reported by the US Federal Trade Commission in 2005 (Dinev, 2006). Previous research has already indicated links between the adoption of online banking and a user’s perception of trust and security (Aladwani (2001) and Suh & Han (2002) were cited by Nilsson & Adams, (2005)). Cameron (2005) also voices fears of loss of credibility and acceptance.

In October 2005 the US Federal Financial Institutions Examinations Council (FFEIC) issued a warning to financial institutions that the use of a password on its own is insufficient for authentication purposes (Federal Reserve System, 2005).

Identifying an authenticating party by contacting a device that belongs to the bone-fide user via an out-of-band channel as described by Wu, Garfinkel and Miller (2003) might address some of the concerns of the FSA.

1.2 Key Aim and Objectives

This thesis and its resulting project have the following main aim:

To propose that an out-of-band self-authentication approach to online identity proof addresses the matters highlighted in section 1.1.

The thesis and project can then be divided into the following 2 objectives:

Describe and build a Proof of Concept prototype of the system similar to the one proposed by Wu, Garfinkel and Miller “Secure Web Authentication with Mobile Phones” (2003) built using the more up-to date paradigms of WAP Push and Web Services.

Specify and build clients in JavaMe and Windows Mobile 6 that will use the above prototype infrastructure to complete the out-of-band authentication loop.

A secondary goal is to find out during this project how a cross-section of users feel about using Information Cards (InfoCards) with and without out-of-band self-authentication.
1.3 Approach and Future Goals
This thesis presents the implementation and evaluation of an authentication process via an identity provider that has an out-of-band leg to a mobile device. Further work would compare and contrast other competing mobile phone technologies when applied to an out-of-band authentication scenario for a web portal.

1.4 Thesis Outline
This thesis opens with a literature review which examines identity, its management and theft; it then reviews authentication systems and mobile device technologies concluding with the main points of interest. The third chapter is a requirement analysis, it examines the topology of the system and some of the technologies used to implement it. Chapter four focuses on the design of the system and outlines the important choices that were made during the project. The fifth chapter discusses the Project’s implementation phase and shows how the parts of the system were coded and how and where the encryption functionality was handled. The evaluation chapter gives details and analysis of a small usability study and decryption performance of a third-party encryption suite in the .Net Compact Framework versus a similar algorithm with no decryption. The Thesis is rounded off with conclusions in the final chapter and proposals for any future work.
2 Literature Review

Introduction
This literature review consists of four separate sections. The first introduces the topic of identity management, a brief history of its origins and then goes on to define identity theft and its various mechanisms; the second focuses on authentication by generally looking at the history of authentication processes and issues surrounding authentication, then discussing in detail current types of Authentication Systems; the third introduces issues in the mobile computing environment and two of the current mobile platforms Windows Mobile 6.x and JavaMe; the fourth and final section makes some conclusions summing up the research and makes some recommendations for the direction the project should take.

2.1 Identity Management
This section outlines a brief history of identity management. Authentication in the context of this project is the proof of identity of a human being. It is necessary to briefly discuss the links between philosophy and identity and to consider the origins of identity management.

2.1.1 History of identity management
This section outlines history of personal identity in Scotland, which provides the thesis with background and context relevant for its intended audience.

Scottish Identity Management (General Register Office for Scotland, 2008)
Identity management in Scotland began in 1551/2 with the Church where the Provincial Council of the Scottish Clergy recommended that a birth and marriage register be kept. In 1574 the General Assembly of the Church of Scotland further recommended that parishes keep a list of all deaths as a part of the Church’s income was derived from the estates of the deceased.

The Registration of Births, Marriages and Deaths Act became law in 1854 outlining that a standardised system of registrations would be undertaken in Scotland. Compulsory registration started on the first of January 1855.

1989 saw the beginning of replacement of paper indices by computerised records with records being made available on-line in 1998.
2.1.2 Identity Theft

Identity theft can characterised by the clandestine procurement of personal information by a third party for malicious use or as the US Federal Trade Commission defines [it]... as “occurring when someone uses your personally identifying information ... without your permission, to commit fraud or other crimes” (Whitman & Mattord, 2009). Identity theft criminals will gather enough information to make transactions on a victims e-commerce account for the criminal’s profit, or the creation of additional e-commerce accounts that only the criminal has access to without the victim’s knowledge. Once an unlawful account has been opened it can be extremely difficult for the victim to find out details of the fraudulent account because of the unwillingness of e-commerce counterparty to share details of the fraudulent applications, which in turn makes it difficult to repudiate ownership of that account (Furnell, 2007).

Mechanisms of Identity Theft

Identity theft can take place via various mechanisms; this section describes some of the mechanisms.

**Dumpster Surfing** is a low-tech method described by Eisenstein (2007) is the physical theft of identity information by rooting through rubbish or stealing physical documents

**Shoulder Surfing** is characterised by (Philpott, 2006) as the lowest tech method of stealing identity by looking over someone’s shoulder to glean vital information that the bone-fide authenticator knows like a pin number or password

**Clandestinely installed client malware** will typically be installed using social engineering on the target identity’s host machine, by enticing the user with free software or media (Gonzales & Majoras, 2007). The malware will then gain identity information whilst on the host machine and send it back to the identity thief.

**Spyware** is defined by Warkentin, Luo & Templeton in their spyware assessment framework as “a client-side software component that monitors the use of client activity and sends the collected data to a remote machine” (2005) that is either installed on the client machine with or without the user’s knowledge. Payton (2006) also agrees that all spyware is not malware and is more commercially motivated. For these reasons this project will use the term Malware to represent autonomous malicious spyware that is used to steal identity for malicious purposes. A brief list of these autonomous malware categories (Payton, 2006) is given below to aid in giving the reader some context.
• **Application Monitoring Tools** are used to glean identity information from email and instant messaging programs.

• **Packet Sniffers** will monitor network traffic from the host searching for identity information such as Bank Account and Credit Card numbers, passwords and pin numbers.

• **Keyloggers** detect the user’s keyboard input at kernel level to glean information.

• **Phishing** is the most common type of social-engineering attack (Gonzales & Majoras, 2007) and can be defined as the practice of sending a spoof communication to an identity target that purports to be from a bone-fide e-commerce institution. The communication will ask the targeted user to provide identity information as part of a verification process that the e-commerce organisation is undertaking.
  
  o Pretext exploits (Gonzales & Majoras, 2007) are where the identity thief contacts the e-commerce institution with the target’s name, and pretends to be the target in order to elicit more identity information.

  o Phone phishing or Voice Over IP Phising “Vishing” (Gonzales & Majoras, 2007) is a relatively recent development which uses telephone calls as a carrier for the spoof communication.

  o Email phishing is the practice of using emails as the carrier for spoof verification messages.

**Web-Spoofing & DNS poisoning** is defined by Dinev (2006) as the practice of creating a hoax web site that closely mimics the real site to extract personal information. DNS poisoning can be described as the hijacking of the target user’s host machine name resolution lookup so that it will point the target identity towards a spoof web site instead of the real site the target wanted to visit.

**User self-compromise** describes the process of a target-identity who disseminates identity information freely. The user might or might not be aware of the issues surrounding identity theft and if in the public-eye may have no security-through-obscurity choice (Leyden, 2001).

• **Social Networking / Blog sites** are highlighted as a danger by Choo & Smith (2007). The posting personal information on a site could be used in a context aware pretext or phishing attack or even to construct the identity for use in the opening of a bogus e-commerce account.

It is important to note that identity theft can occur by a variety of mechanisms and there are numerous mechanisms. One of the main points made by Gonzales and Majoras (2007) in their Strategic plan is that a single common identifier like a Social Security should not be used for identification one may ask though, how is a user to repudiate false accounts as described in Furnell’s (2007) case study if there is no universal identifier; a social security number could be
argued to be a better identifier than most as Governments may well be perceived to be enthusiastic for their citizens to pay tax.

2.2 Online authentication processes

An authentication process in the context of this thesis can be defined as a remote user providing information to a remote service via a digital network to gain access to a remote resource. The authentication process can be broken into two mechanisms (Nilsson, Adams, & Herd, 2005) the identification of an authenticated session and the verification of the user’s identity; this project focuses on the verification of the user’s identity. Information or proof (RSA Security) that is used to validate a user's identity is often called an ‘authentication factor’ or ‘authentication method’. It is generally agreed that authentication factors fall into 3 categories (Summers, 1997) and (Federal Reserve System, 2005):

1. Possession: a proof that the user possesses a unique object at the time authentication takes place. E.g. a card, token or device.

2. Knowledge: the system challenges the user for a specific piece of knowledge that only they will know in the given context. E.g. a password or passphrase.

3. State of Being: something the user is or does at the time of authentication. E.g. Biometrics: Fingerprint, Iris Scan, Keystroke pattern.

It is possible to define a further 2 categories although it could be argued that these are encompassed by the 3rd category.

4. Location: the user is at the expected location at the time of authentication as demonstrated when they physically present themselves at the airport check-in. This is true whenever it is required to produce photographic identity proof.

5. State based: the user’s authentication attempt is at an expected time. If identity proof is linked to a stream of consciousness introduced in section 1.1 then the user’s attempt is expected by themselves or even by another e.g. the Check In clerk at an Airport waiting for a passport to be produced; this introduces the concept of awaiting authentication as a state of being is therefore state-based.

Generally speaking the more factors used in the authentication process, the stronger the security.

Two of the fundamental problems as we shall see are the digitisation of information and the inability of Internet protocols to verify a user’s location.
Single-factor authentication

Single-factor authentication is the process of using one type of authentication proof to validate a user’s identity. Because of easy implementation a username and password and or passphrase has until recently been widely used for Internet Banking and e-commerce but is now not sufficient (Federal Reserve System, 2005) due to phishing, pharming and malware.

There seems to be a consensus of opinion that single-factor authentication is a thing of the past (Information Age, 2006) and (Computer Fraud & Security, 2003).

Multi-factor authentication

Multi-factor authentication is the process of using more than one type of authentication method, to validate a user’s identity. Multi-factor authentication is thought to more difficult to compromise (Beaumier, 2006)

2.2.1 Authentication systems

Password / Passphrase / Memorable date

This knowledge based authentication method is still the most prevalent form of authentication today and is used in almost all single-factor authentication systems and as a preliminary stage for a multi-factor authentication system.

The user is prompted to enter a username which is used to identify a valid system user and is also prompted to enter a password which is intended to prove the user’s identity. In terms of evaluation threat of malware sitting quietly on a user’s machine stealing passwords and passphrases (Greiner, 2006) has caused the US FFIEC to issue guidelines (2005) to its members to consider additional authentication methods. The Home Office has stated that (Nicholas, Kershaw, & Walker, 2006 /2007) the largest type of card fraud between 2005 and 2006 came from non face-to-face transactions over the phone or the internet, it is unclear what percentage relates to details being obtained by malware but the 16% increase over the course of a year could be linked to the development of new types of malware identified by Greiner (2006) and Kabay (2005)

Digital Certificates / Public Key Infrastructure (PKI)

Digital Certificates can be classified as possession based authentication factor category. With reference to the mechanism, a user provides their Digital Certificate to the service as proof they are the valid user. Digital certificates use public and private keys to validate a user’s identity using a digital signature. The most common solution (Summers, 1997) is for a user to register their public key with a trusted Certificate Authority (CA), which then distributes the public keys on the user’s behalf.
Using PKI and the issuance of Digital Certificates to validate a user’s identity is in essence a good idea but there are many reasons why its adoption hasn’t been as successful and widespread as the IT community hoped it would. Ellison and Schneier (2000) come up with 10 compelling risks of PKI from “who do we trust, and for what?” to more importantly “How did the CA identify the certificate holder”. Backhouse, Hsu, Tseng and Baptista (2005) argue that failure of the public key infrastructure to achieve widespread adoption by users and services lies in the users uncertainty in the quality of the authentication proof.

**Biometrics**

Harris & Yen (2002) quote Ashbourn (no ref, 2000) in defining biometrics as “a measurable physiological and/or behavioural trait that can be captured and subsequently compared with another instance at the time of verification”. These traits can be described as fingerprints, palm prints, retina scans key stroke and voice patterns. In Evaluation of bio-metrics Harris and Yen argue that momentum behind biometric authentication systems and the technological lead the US has in biometric systems will outweigh the obstacles of social acceptability and inclusivity of those systems highlighted by Braz and Robert (2006). As previously highlighted single factor authentication systems have their critics and it would seem highly inadvisable to rely on a single factor system based on biometrics as those characteristics cannot be as easily changed once that data is compromised

**Tokens / Smart cards**

Financial Institutions have in the past issued RSA’s SecureID® token (see Figure 1) to their employees who issue bank to bank cash movements; Bank of Scotland is now issuing these tokens to their internet banking customers. Tokens or smart cards are an example of a one-time password (Bailey, Vongsathorn, Kapadia, Masone, & Smith, 2007). In evaluation terms Bailey *et al* point out secure tokens and smart cards protect against a lost password but do not protect against a fully hijacked session.
Pre-shared Keys
A pre-shared key is one of the oldest forms of encrypted communication and has certainly been around since the Roman Empire (Whitman & Mattord, 2009). A key is generated and is kept secret between parties this key is then used to encrypt communications between those parties. In recent history key generators have been used with encryption algorithms like DES, 3DES and most recently AES. On examination the DES 56 bit key encryption has now been proven to be unsafe (Whitman & Mattord, 2009). Whitman & Mattord tell us that AES 256 bit key encryption is the defacto standard of the US Federal Government (FIPS). In a distributed system pre-shared keys suffer from one fundamental problem, that of how to exchange the key through a secured channel.

Trusted Identity Provider
Cameron & Jones (Cameron, The Laws of Identity, 2005) outline their vision of an internet-wide identity meta-system based on trusted identity providers in their article “Design Rationale behind the Identity Metasystem Architecture”. The Identity meta-system in Figure 2 essentially borrows its architecture from Kerberos (Buchanan, Enhanced Software Security, 2007) where a relying party (an e-commerce institution in the scope of this report) trusts the identity of the authenticating user because the user has been granted an access token by an Identity provider the user is registered with. Chappell (2006) describes the Identity metasystem as a system of systems focused on Identity. He further points out to make the Identity metasystem work, “requires cooperation [and] ... No single organisation can unilaterally impose a solution”.

Figure 1: RSA SecurID® USB Dongle (RSA SecurID, 2008)
Microsoft has implemented their client identity selector under the name Windows CardSpace (formerly known as InfoCard) that uses cards known as InfoCards. The OpenID foundation has also implemented a similar federated identity meta-system with OpenID cards, their Corporate Board Members represent Google, IBM, Microsoft, VeriSign and Yahoo!. In the OpenID model a Diffie-Hellman key exchange takes place between the Relying Party and the Identity Provider. On assessment the Trusted Identity Provider (TIP) system has many strengths. It conforms to all of Cameron’s seven laws of identity foremost of which by protecting a user’s digital identity information which is held securely by the Identity provider. Microsoft’s CardSpace however does not prevent hijacking of a session via inadvertent access granting and impersonation (Rowley, 2007). In an email from Cameron (Cameron, Error in Blog, 2008) himself acknowledges that impersonation access could take place if the public key-pair/password backed card is backed up and the backup falls into the wrong hands given the UK Government’s propensity to lose unencrypted data and their own acknowledgement that it can’t be prevented this is significant weakness (Hutt, 2008). If the card is backed by a hard-token like RSA SecurID it seems to be a much safer system. Cameron also points out another potential issue where users purposely give their cards away to masquerading parties who then qualify against restrictions.
based on the user’s identity and not theirs, the example given is an underage drinker who is allowed to buy alcohol over the web based on the user’s age.

Out-of-band authentication gives users something that they have not had before: The ability to see how many bogus authentication attempts are being made to their online resources.

In Kim Cameron’s email (Cameron, Error in Blog, 2008) he highlights that users may purposely give away their managed cards to aid illegitimate users in their procurement of restricted goods and services. An out-of-band self-authentication mechanism might add a degree of hardship to the user who has divulged their managed cards to a wide audience as their mobile device and their time will be taken up with authenticating these attempts.

2.3 Mobile device technologies

Mobile devices in the context of this project shall be considered to be handheld devices that have Short Message Service (SMS) and Internet connectivity capability via the cellular phone network. This section will discuss the types of mobile networks and connectivity available, and two of the client device platforms in the market today, JavaMe and Windows Mobile 6.

2.3.1 Mobile cellular networks

**Brief History in terms of Generations**

Mobile transmission technologies are often and widely described in terms of their generational acronym; from first generation (1G) through to the third and current generation (3G).

**1G analogue transmission** was first introduced the 1980s and was primarily intended for voice telephony (B’Far, 2005) by the start of the 1990s 2G digital transmission had broken on to the mobile telephony scene and accounted for the bulk deployments of voice and data services in Europe and the United States (B’Far, 2005). The categorisation 2G actually includes but is not limited to the following three technologies of Time Division Multiple Access (TDMA), Global System for Mobile (GSM) and Code Division Multiple Access (CDMA).

The **2.5G label** is applied to transition technologies that are essentially extensions to the 2G transmission carriers that enable faster data transfer an example is General Packet Radio Services (GPRS) that allows a data transfer rate of up to 100Kbps.

Europe seems to have settled on Universal Mobile Telecommunications System (UMTS) based on W-CDMA as the **3G transmission** technology; however the United States has the more fractured choice between WCDMA, TD-CDMA and CDMA2000.
Devices available for purchase in the European high street today will either come enabled with both GPRS and UMTS or with just UMTS.

2.3.2 Internet Connectivity

WAP

Wireless Access Protocol (WAP) according to B’Far (2005), Gavalas and Economou (2007) is the single most pervasive framework used for building most mobile applications available today. WAP was originally founded by Nokia, Ericsson, Motorola and Unwired Planet in 1997 (Orange). The WAP 1.0 stack was designed with low bandwidth mobile connections in mind and supports its own Wireless Mark-up Language (WML) it is independent of the underlying Mobile network technology. Initially designed to be as pervasive for mobile applications as HTTP has been for the Web (B’Far, 2005) WAP mirrors the client-server architecture of browser and a web server. The initial success of WAP 1.0 was marred by its inability to meet expectations but with the advent of WAP 2.0 it seems to have overcome its critics. WAP protocols (Figure 17 & Figure 19 in Appendix C) are designed to operate over a variety of different bearer services, including SMS, circuit-switched data and packet data (m-indya). WAP can be useful technology when combined with the Push Access Protocol (PAP) which enables actions to be pushed to the

Figure 3: Wireless Web technology landscape (Gavalas & Economou, 2007)
mobile device.

**NTT DoCoMo's i-Mode**

i-Mode is a closed and proprietary system that was launched in 1999 as a direct competitor to WAP, the i-Mode gateway or “i-Mode center” is under the control of NTT DoCoMo (Roxburgh, 2006). Unlike WAP which uses a variety of bearers, i-Mode uses a Personal Digital Cellular Packet (PDC-P) ... over the existing ... network used for voice traffic (Lakhani, 2001). i-Mode also uses its own mark-up Language of c-HTML which is a superset of HTML and is only available to i-Mode compliant phones.

In evaluating the wireless web, Gavalas and Economou’s (2007) characteristics for wider adoption and growth of wireless technologies are defined as follows:

1.) Ubiquity and convenience: Mobile devices satisfy the need for real-time communication with no time and place constraints

2.) Positioning: using technologies like Global Positioning System (GPS) users may receive and access information and services specific to their location (reference to Varshney, 2003).

3.) Personalisation: handheld devices are typically operated by a s single user, thereby enabling the provision of personalised services by wireless web portals (references to Ho & Kwok, 2003; Varshney et al, 2004, Mahatanakoon, Wen and Lim, 2006)

**2.3.3 Windows Mobile 6.1**

Microsoft entered the mobile telephony market with Microsoft Smart Phone 2002 (B'Far, 2005), Microsoft Smart Phone and Pocket PC products were merged into the Windows Mobile platform for the release of Windows Mobile 2003. Currently Windows Mobile has 140 devices in the marketplace on over through 125 mobile operators world-wide (Microsoft, 2007). Microsoft provides the following description of Windows Mobile:

“Windows Mobile is a platform for mobile devices based on Windows Embedded CE” (Microsoft, 2008)

.Net Compact Framework
Microsoft describes the .Net Compact Framework described in Figure 4 as: “a subset of the .NET Framework ... [containing] features exclusively designed for the .NET Compact Framework.” (Microsoft, 2008).

![Figure 4: .Net Compact Framework (Microsoft, 2008)](image)

2.3.4 Java Micro Edition (JavaMe)
JavaMe is a platform developed by SUN Microsystems and was formerly known as J2Me. JavaMe in the context of this project refers to JavaMe Connected, Limited Device Configuration (CLDC). CLDC as highlighted in Figure 5 has been developed for devices that have a running memory between 32 and 512 KB and runs on the K-Virtual Machine (KVM) (B'Far, 2005). Sun tell us that the JavaMe technology is based on three elements (Sun Microsystems, 2008):

1. a configuration provides the most basic set of libraries and virtual machine capabilities for a broad range of devices.

2. a profile is a set of APIs that support a narrower range of devices, and

3. an optional package is a set of technology-specific APIs.
The following statement from Gavalas and Economou which serves equally well for the .Net Compact Framework as it does for JavaMe, suggests that mobile client applications designed for a specific task suit the context of a mobile environment much better than a generic browser, this development strategy has been borne out by Google’s development of its JavaMe Google Mail application:

“J2Me provides a vehicle for creating complex applications with a higher degree of interactivity compared to their (browser based) WAP or i-Mode counterparts. Independence ... from the underlying transport protocol is expected to lead to a synergy of the J2Me and WAP/i-Mode worlds” (Gavalas & Economou, 2007)

The vital element of this project is that the user is part of the authentication process. It will be necessary to justify self-authentication and in doing so there is a need to re-examine the philosophical issues and practical issues.
2.4 Conclusions
The intentions of this thesis and its related project are to implement a Secure Token Service and an arbitrary web-portal that requires tokens from this service via a Microsoft Card-Space Identity Selector client. It is hoped that this prototype system will frame of reference within which the online security community may re-visualise authentication systems and refocus on some of the fundamental questions that relate to identity proof.

2.4.1 The Identity Theft problem
It has been seen in this literature review that identity theft is a serious and increasing problem; opportunities for and reward via theft by criminals are increasing whilst the risk of being caught is decreasing because of distributed technologies.

2.4.2 The March toward Managed Cards
It has become all too apparent during the research for the literature review that Managed Cards and trusted Identity Providers are a credible solution to the failure of Digital Certificates. It has also become abundantly clear that large organisations like Microsoft, IBM and Google are pushing towards the accountability of users and services in the internet via these Managed Cards and trusted Identity Providers.

Questions still need to be asked specifically of the relationships between the Relying Parties and the Identity Providers and of the credibility of these Providers both in the eyes of the Relying Parties and in the eyes of the trusting Users. OpenID themselves recognise these issues and state on their website:

“You get to choose the OpenID Provider that best meets your needs and most importantly that you trust” (OpenID Foundation, 2008)

Much of this debate is out of the scope of this project, however the use of self-authentication via an out-of-band loop might help users select and trust Identity Providers.

2.4.3 Mobile Technologies
Activating a mobile client program using an SMS message is a powerful technique for distributed systems that have a mobile element. The mobile component is by its nature is resource starved and cannot always be connected to the system and for this reason the SMS wake-up capability is so powerful.
3 Requirements Analysis

3.1 Introduction
This Chapter has two main points of analysis which were derived from Figure 6 that shows how Wu, Garfinkel & Miller’s (2003) proposed “Secure Web Authentication with Mobile Phones” system fits into the managed card infrastructure.


2. Simple Message Service Centres.

Figure 6: Out-of-band self-authentication as part of Identity Provision
The identity metasystem is a distributed system (Chappell, 2006) that facilitates Identity proof through the exchange of security tokens and each token represents information about a digital identity. Trusted Identity Providers (TIPs) and the clients that use them are at the heart of this metasystem and illustrate the Client-Server paradigm, the TIPs can be installed on any platform using any type of Security Token. InfoCards can be used for authentication to Windows Live.

### 3.2 Identity metasystem components

The Identity metasystem comprises of three main components:

**A Relying Party** is a remote service that requires accountability for users and services that access it. In the context of this thesis the relying party is a standard web server running in IIS 7.0 on a Windows Server 2008 that implements the following requirements list (Chappell, 2006)

- Accepts security tokens.
- Defines its Security Policy.
- Makes its certificate available.

**The Identity Provider** satisfies the following four requirements (Chappell, 2006):

- Create Information Cards (InfoCards) that are compatible with the Microsoft-defined card format, and provide a delivery mechanism for these cards.
- Implement a Security Token Service (STS) as defined in the WS-Trust specification that can issue tokens in any format.
- Define a Security Policy using WS-Security Policy, and allow access to this policy using WS-MetaDataExchange.
• Indicate how the policy requests that security tokens be authenticated. There are currently four supported options:
  
  o Username / password
  
  o Keberos Tickets
  
  o X.509 v3 certificates
  
  o SAML security tokens created by a self-issued identity provider

**CardSpace Client Identity Selector.** The Windows CardSpace selector is a client application that is bundled with Microsoft Vista and is available in Windows XP as update. It allows users to choose InfoCards that are signed by Identity Providers based on the Relying Party’s token requirements.

Vittorio (An Identity Provider and its STS: writing a custom STS with the October Beta of the Geneva Framework, 2008) elaborates on these three main elements to come up with a total of six components in his article.

### 3.3 Identity Provider Components

The Secure Token Server (STS) in this thesis is a server running a WCF Web Service application in IIS 7.0. It receives security token requests from the CardSpace selector, and checks the identity of the card against the registered user store, the Security Token Service then grants a Security Assertion Mark-up Language (SAML) token back to the CardSpace client.

Ideally an Identity provider should have an **attributes store** for users this could be an Active Directory which is then queried by the Secure Token Service and on successful validation, the service returns a secure token back to the CardSpace client.
3.4 Simple Message Service Centre (SMSC)

Looking again at Figure 6 and taking into account Figure 7, an out-of-band loop can be added to the Identity metasystem by getting the Secure Token Service to send an authentication verification request to the user’s mobile client via an SMSC. There are many SMSC’s operating in the market, it is important to select one that provides easy integration with the STS’s technology. Upon conducting research Esendex is an SMSC based in the UK which was found via “The Code Project” website. Sample C# code is made available in Bird’s article (2004) that accesses Esendex’s API via a Web Service.

Figure 7: Out-of-band loop in focus
3.5 Communication Protocols
Web Service protocols are discussed in Appendix E.

A custom encryption solution for OTA communication based on Authenticating the message and sender (Buchanan, 2007) is needed to ensure the Confidentiality, Integrity and Assurance (CIA) of the authentication authorisation transactions that are communicated wirelessly between the Federated Identity Infrastructure and the Mobile Clients – The custom solution is an example of end-to-end security as the encryption takes place in the Application Layer of the OSI Model.

3.6 .Net 3.5x Windows Communication Foundation (WCF)
Windows Communication Foundation is a service-oriented framework designed for distributed computing solutions that uses the .Net Common Language Runtime (CLR). WCF uses the abstract concept of endpoints which allows services to expose use a variety of different transport mechanisms including but not restricted to:

- **NetTcpBinding** SOAP messages using TCP
- **BasicHttpBinding** Basic SOAP transport based on WS-I Basic Profile
- **WSHttpBinding** Secure SOAP transport based on WS-* protocols and in particular to this Thesis it is noted that it supports WS-Security 1.1 and WS-Trust (Microsoft, 2007)
- **WebHttpBinding** A RESTful way accessing resources using HTTP GET and POST methods.

3.7 Windows Mobile 6
3.7.1 Development Tools
Visual Studio 2008 is the Integrated Development Environment advocated by Microsoft to program Windows Mobile 6 .Net Compact Framework applications. It comes packaged with two emulators that aid development of mobile applications without having a physical device to hand:

1. a Device Emulator which enables testing and debugging of applications that run on the mobile device
2. a Cellular Emulator which emulates a GPRS / 3G cellular network and the services that are provided by mobile phone operators.
3.7.2 Application Programmers Interface (API)

Figure 20 in Appendix D shows an overview of the Windows Mobile SDK described as a functionality Map. There are two styles of development on the Windows Mobile platform using either unmanaged or managed code.

Unmanaged code is programmed using Visual C++ with the API reference it allows developers to communicate directly with the hardware of the mobile device. A Developer may choose this development approach if they need the application to run quickly and not take up many resources, however development using this environment is non-trivial and can easily cause device crashes through memory errors.

Managed code is programmed using either Visual Basic or Visual C# and is easier to program with as it operates in the Common Language Runtime (CLR) of the device which handles memory using a garbage collector. It is worth noting that not all of the device’s hardware can be accessed using managed code. A design decision was taken given lack of experience with unmanaged code and Visual Basic to implement everything in C#.

Application activation by SMS

Receiving WAP Push Binary messages are only accessible using unmanaged code, but it is possible to open up an application by checking ordinary SMS messages for criteria. The latter approach to application ‘wake up’ was chosen.

Encryption

The Windows Mobile platform offers both symmetric and asymmetric encryption through the CryptoAPI, the symmetric Algorithms in Table 1: Windows Mobile Symmetric Encryption (Microsoft, 2008). The current version of Windows Mobile does not support AES symmetric encryption which is the encryption algorithm recommended to protect US National Security Systems (The Committee on National Security Systems, 2003). Another third party API called Bouncy Castle is available for C# this API is also available for JavaMe CLDC. BouncyCastle offers the following support:

- AES 256 bit Encryption
- SHA 256 bit Hashing
- RSA 1024 bit Public private key encryptions
Choosing BouncyCastle as a Cryptographic Service Provider lends the Thesis more security focused credibility with an AES 256 bit encryption and will enable a proper comparison between the two Mobile Client platforms.

Table 1: Windows Mobile Symmetric Encryption (Microsoft, 2008)

<table>
<thead>
<tr>
<th>Supported stream-based encryption algorithms</th>
<th>Supported cipher encryption algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC2</td>
<td>DES</td>
</tr>
<tr>
<td>RC4</td>
<td>3DES</td>
</tr>
</tbody>
</table>

User Interface

Development of a .Net Compact Framework application for a Windows Mobile device is similar to the development of a Windows application using the inbuilt designer in Visual Studio. The main application is started when the initial form loads, subsequent forms can be instantiated from within the original parent form and run as child forms.

Replying to the Identity Provider

Referencing the Phone Authorisation web service is a case of adding a Web Reference to the Visual Studio project and specifying the location of the web service’s description language document (WSDL).

Deployment

The Windows Mobile 6 SDK can be deployed to either an emulator or a device attached to the machine.

3.8 JavaMe

3.8.1 Development Tools

Applications for the JavaMe platform are more disparate and are very much device manufacturer led, with manufacturers providing emulator and connection tools based on Sun’s Wireless toolkit. Device manufacturer’s also give advice on which IDE’s to use. In this project, development was governed by having access to a Sony-Ericsson K810i Mobile phone. Sony-Ericsson have a development portal called Developer World that has all of the tools necessary to develop applications for the JavaMe platform for Sony-Ericsson phones. The Sony-Ericsson toolkit includes NetBeans with the Sony-Ericsson MIDP CLDC profiles pre-installed.

The portal also has a number of video and text based tutorials to help developers gain an understanding of how to develop for the JavaMe platform for Sony-Ericsson devices.
3.8.2 API

Application Activation by SMS
Applications can be activated by a WAP Push binary message (B'Far, 2005). These are very limited in length and typically open up the mobile device’s browser automatically directing the browser to the URL which is contained within the body of the Push Message. A custom application can be easily configured in NetBeans to listen on a port number. When the binary message is sent with its port number, the application is activated. It is important to note that the Listening port handler is part of the devices’ proprietary operating system rather than being part of JavaMe.

Encryption
As previously discussed in section 3.7.2, JavaMe applications use a third-party API called Bouncy Castle, which has all the encryption providers and algorithms to implement this Thesis’ Project.

User Interface
NetBeans uses a Visual Midlet Designer which has a concept of application flow which links the forms needed to run the application together.

Replying to the Identity Provider
Like Visual Studio, a web service is set up using a Wizard which asks for the URL of the service’s WSDL document.

Deployment
Sony Ericsson have their own deployment application to the Mobile Device, unfortunately I encountered errors using this application and was only able to transfer the JAD and JAR files manually using a file explorer.
3.9 Conclusion

WCF provides enough functionality to implement the Federated Identity Management Infrastructure. The following elements and their functionality have been realised through the conducted analysis:

**WCF Secure Token Server that will:**

Accept Token Requests; Verify Registration of a User; Send Authorisation Requests to an Authorisation Service and await responses; Accept Authorisation Responses and Give out Tokens accordingly.

**An Authorisation Service that:**

Accepts authorisation attempt request from an STS; Sends an application activating SMS to a Mobile Client; Can provide encrypted attempt data to a Mobile Client; Receives authorisation attempt responses from a Mobile Client; Sends authorisation responses back to an STS.

**A Mobile Client that will:**

Receive incoming SMS-es from an STS and activate an application to process attempt receipts and responses; Fetch attempt data from an Identity Provider server; Decrypt attempt data; Send a response back to the STS.
4 Design

4.1 Introduction

This chapter deals with the phase of the project in which decisions and compromises had to be made in order to realise the project objectives as fully as possible and within the existing time and cost constraints. It was necessary to choose an appropriate architecture so that the hardware and software components could be linked together at the implementation stage balancing performance, availability and cost amongst other considerations. A major consideration was the time available to put the design into effect and verify that it was in fact performing as planned.

Proving digital identity across a distributed system can be seen non-trivial challenge set against a constantly evolving human and technological backdrop. Thoughtful design is the key to developing sets of “Bronze Bullets” that address remote Identity Theft as “Bulletproof security does not exist” (Sequeira, 2003). One of the desired outcomes of this thesis is the design and implementation of a self-authentication solution that adds considerable complexity to the process of Identity Masquerading without impacting too much on the process of Authentication by the User. Designing in self-authentication as highlighted by Dean (2006) might be one of these “Bronze Bullets”.

To create the infrastructure described in Chapter 3 it becomes apparent looking at the two network topologies in Figure 6 and Figure 7 that a great deal of this project involves design and implementation of the authentication infrastructure, even though the implementation of the project is focused on the two client technologies of the .Net Compact Framework and JavaMe CLDC.

4.2 High-level Abstraction

Analysis of Figure 6 and Figure 7 and earlier analysis have shown that the system has a distributed Client-Server architecture and gives rise to the conceptual design for the system Figure 8. The Security Token Server is accessed using WCF Web Services from a Windows CardSpace Identity Selector on the non-mobile client. The Authorisation Service is accessed using WCF Web Services from a custom application based on the mobile client and from the
Secure Token Service. It will be necessary to secure the communication between the mobile client and the Authorisation Service as this is over-the-air OTA and could be listened in upon both at the Simple Message Service Centre and at the Mobile Operator.

Figure 8: Identity Provider Abstraction

4.3 Persona / Storyboard

Personas and Storyboards can be a useful part of the design process helping (Benyon, Turner, & Turner, 2005) designers focus on the stakeholders in the proposed system. The following narrative illustrates how it is envisaged that the proposed system would operate in a real-life situation. It is based on an imaginary person, Arthur.

Arthur is retired and in his late sixties, he has a number of online e-commerce and bank accounts and regularly makes transactions from his home computer over the internet. Although Arthur is non-technical, he is cautious about internet security and has installed Antivirus Software and a Software Firewall on his computer; both of which were produced by a reputable company and were bought directly from their site. Arthur always tries to keep his operating system up-to-date and Arthur’s home computer accesses the internet from behind a home wireless router.
Recently Arthur has created an InfoCard digital identity because, after reading several articles about identity theft, he was worried about the security of passwords. Arthur registered his digital identity in person at the office of an agent of a reputable Identity Provider whose operations and procedures are regularly assessed for their quality and efficiency. Arthur's identity provider allows Arthur to self-authenticate using his Mobile Phone because whilst in their office the agent installed a secure application upon it. Whenever Arthur goes to log on to his online e-commerce or bank accounts with his InfoCard, his Mobile Phone starts an application asking him whether he wants to verify the access attempt. On one occasion Arthur was not accessing any of his accounts and his Mobile Phone registered an unauthorised access attempt that he was able to report directly back to the Identity Provider using his Mobile Phone.

4.4 Design Considerations

As this system is a Proof of Concept Prototype and not a commercial system, many assumptions and tradeoffs were made before arrival at the final design decision.

4.4.1 Assumptions

Infrastructure

It is assumed that this system does not hide failure as described in the Tanenbaum & Van Steen (2007) Distributed Systems Transparency model. Due to limited resources, the system does not implement replication and redundancy.

Server Security during development, testing and demonstration was minimal as this was not a focus of the implemented project – It was therefore assumed that no malicious attacks would take place on the Identity Provider and its associated registered user/services store.

Mobile Client

False authorisation of an authentication attempt might arise either from the mobile device falling into the wrong hands or the mobile user being subjected to duress. Howie (2008) suggests the use of a pin number and a duress pin/code described by Clark and Hengartner (Clark & Hengartner, 2008) to safeguard the device's private encryption key and the release of the authorisation-response. On analysis it is possible to implement this security screen using voice recognition through a Voice User Interface (VUI) which offers clear advantages to the mobile context (B'Far, 2005). The inclusion of this screening functionality would have been ideal as part of the design however due to time restrictions it is assumed that the person with the device is the correct person and that they are under no duress.
It was assumed that the Mobile Client and the Identity Provider had already exchanged their public keys as described in the Storyboard. This assumption lends itself to a scenario where the user has identified them self (and presented their mobile device) in person to a trusted agent of the Identity Provider who has in turn verified their identity – this would seem to be a robust model that although not infallible and satisfies the issues put forward by Ellison and Schneier in their 8th risk (Ellison & Schneier, 2000).

Generic user interfaces are an important aspect of application design for mobile devices “as there is such a wide array of end clients for mobile applications” (B'Far, 2005). Again for the sake of simplicity it is assumed that there is only one type of end device per platform and that the user interface has been developed specifically for it.

4.4.2 Design Tradeoffs

The following design decision tradeoffs were made to aid in simplifying the implementation of this project.

**Microsoft products** were chosen for the Relying Party, Web Client and Identity provider as these were critical to the success of the project, this choice was to aid:

- Interoperability.
- Ease of implementation during the time-frame because of previous experience with Visual Studio, C# and .Net MSDN documentation.

**Web Services** were chosen over a socket based or remote object based system as the author’s familiarity with Web Services at the time of development was greater than other methods.

The ideal implementation of this system would use **secure http sessions** to connect to the Web Services, time did not allow implementation.

A deferred asynchronous programming pattern was chosen over a more robust implementation using **Windows Workflow (WWF)** for rapid development as time did not allow the acquisition of sufficient knowledge to implement the Identity Provider using WWF.

**Interoperability issues** between the JavaMe Client and the Authorisation Service caused the following design tradeoffs:
• An intermediate .Net Framework 2.0 ASMX web service that sits as an interface between the WCF Attempt Manager web-service and the Mobile Client for Authorisation responses.

• Attempt detail retrieval by the Mobile Client is done via a WCF Restful publishing of XML because the length of string allowed in the Push Message sent via the SMS Centre is not sufficient for all the attempt details.

4.5 Design Decisions

4.5.1 Infrastructure Model
Looking at the abstraction in Figure 8 it is apparent that it is a distributed metasystem and should appear to the User to be a single unified system (Tanenbaum & Van Steen, 2007) to achieve this goal the infrastructure has to be developed with future scale in mind. For scalability reasons, the Secure Token Server and the Authorisation Service should be located on separate machines Fig V shows the hardware components that make up the meta-system.

Figure 9: Hardware Infrastructure model
Identity Provider
The Identity Provider can be thought of as a software interface that obscures many different types of implementation that might be contained with it and protocols that it may use. The CardSpace™ client interacts with a Secure Token Server (STS) which verifies that the user is registered against an information store of registered users/services (e.g. a X509 Certificate Store in the form of Microsoft Exchange Server) as the type of implementation is not the focus of this project, it is sufficient to arbitrarily choose one.

Authorisation Service
In this Thesis the Identity provider masks the Authorisation Service seen in Figure 8. The Authorisation Service manages authentication attempt notifications sent to the verifying party’s mobile phone, and their responses.

The heart of the Authorisation Service (Figure 10) is an attempt manager that uses deferred-asynchronous messaging to communicate with the Mobile Client.
It will be necessary to persist the authentication attempts and for rapid development reasons, this persistence will be a store of XML files.

The Authorisation Service for convenience was split into three separate web services consisting of: a WCF web service that uses the WS-* profile, responsible for initiating, managing and ending authorisation attempts; an ASMX (ASP.NET 2.0) web service configured to use the WS-I Basic Profile that acts as an interface between the STS, Mobile client and the WCF Service; and a RESTful Web Service, that provides encrypted attempt XML data at a URL which the mobile devices fetches upon receipt of the SMS message. Figure 13, Figure 12 and Figure 14 in Appendix B show the WCF, ASMX and Restful web service class diagrams respectively.
Web Client
The Client will also have to implement a CardSpace™ client. According to Dorrans (Interview at Developer Day, 2008) it maybe that a custom CardSpace™ client would have to be built in reality as the out-of-band loop may cause the authentication attempt to time-out, if the user is in an area of non-cellular network coverage.

4.5.2 Simple Message Service Centre (SMSC)
In order to send an SMS that will initiate the Mobile Client application. It will be necessary to create an account at a Simple Message Service Centre provider. A number of these providers have API’s that can be used to send SMS messages. The selection of an SMSC is arbitrary and does not have any impact on the outcome of this project.

4.6 Mobile Client
To complete the out-of-band loop, the client will have to undertake the following tasks:

1. Receive an SMS message WAPPush or otherwise that initiates the client application

2. Receive information in a secure manner regarding the authentication attempt

3. Give the user the option to accept or deny the attempt

4. Send the user’s response in a secure manner back to the Identity Provider so it can release the token.

4.6.1 Security
In order to achieve confidentiality, integrity and availability (CIA) in the over-the-air (OTA) section of the system it was decided for brevity of project implementation to employ a custom encryption scheme to handle communication between the Server and the Mobile Clients. The scheme selected is based on a scheme for authenticating the sender and the message (Buchanan, Authenticating the sender, 2007) which uses Symmetric session keys encrypted with the Mobile devices public key to encrypt the XML element content that is fetched from the server by the Mobile Client. The data is also signed with a one-way hash encrypted with the Server’s Private Key. Referring to the Storyboard, it is assumed that Identity Provider and the Mobile client have exchanged keys face-to-face. The Mobile Client’s Class Diagram can be seen in Figure (X) in Appendix B.
5 Implementation

5.1 Introduction
This chapter explains the general stages that were employed when implementing the designs discussed in Chapter 4; it is divided into 2 sections. The infrastructure section discusses the hardware and software used to implement Federated Identity with an out-of-band loop discussing both the Relying Party and the Identity Provider; the second section explains implementation of the two different Mobile Clients. Finally this chapter is concluded by a summary of prototype weaknesses.

5.2 Infrastructure
This section is categorised into the two sub-sections of Hardware and Software which will hopefully aid the reader in visualising the Implementation process. During analysis of the federated identity design (Figure 8) it became clear that several servers are required in the implementation phase; this impacts on the technology choices and Virtual Server technology was used to fulfil the requirement, and facilitated in the backing-up process.

5.2.1 Hardware
In total seven servers were implemented (the application servers are discussed in Section 5.1.2):

<table>
<thead>
<tr>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory Domain Controller</td>
</tr>
<tr>
<td>Microsoft Exchange Server</td>
</tr>
<tr>
<td>Secure Token Server</td>
</tr>
<tr>
<td>Authorisation Service</td>
</tr>
<tr>
<td>Relying Party</td>
</tr>
<tr>
<td>Host Operating System</td>
</tr>
<tr>
<td>Development Server</td>
</tr>
</tbody>
</table>

Table 2: Implemented Servers
A host with an abundant supply of Memory was needed to accommodate all these application servers which gave rise to the following technology requirements:

- **6GB of RAM** was allocated as follows:

<table>
<thead>
<tr>
<th>Machine Responsibility</th>
<th>RAM Allocated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorisation Service</td>
<td>1GB</td>
</tr>
<tr>
<td>Secure Token Service</td>
<td>1GB</td>
</tr>
<tr>
<td>Relying Party</td>
<td>512 MB</td>
</tr>
<tr>
<td>Domain Controller</td>
<td>512 MB</td>
</tr>
<tr>
<td>Host Operating System</td>
<td>1.5 GB</td>
</tr>
<tr>
<td>Development Server / Test Client</td>
<td>1 GB</td>
</tr>
<tr>
<td>Exchange Server</td>
<td>512 MB</td>
</tr>
</tbody>
</table>

Table 3: RAM Allocation between implemented servers

A **64bit Processor**, (AMD dual core) was chosen for Memory recognition purposes by the host operating system as 32 bit operating systems can only recognise up to 4GB of RAM.

2 **Internal Hard Disks** of 80GB and 250GB were selected to run the host and store the Virtual Machine files respectively. The larger disk kept the Virtual Machine files which were approximately 20GB in size, the smaller disk was responsible for storing the host operating system.

An **External Hard Disk** of 500GB was used to save backups of the Virtual Machine files.

A **home gateway/router** was used as a DHCP server and allowed hosting using the DynDNS service.
5.2.2 Software
Subsection 5.1.2 discusses the project’s infrastructure implementation under the two categories of Platforms used and Applications installed.

Platforms
Microsoft Windows Server 2003 R2 (Win2k3) 64 bit (x64) Enterprise Edition (EE) was installed to host the virtual machines on a hypervisor distributed by Sun Microsystems called VirtualBox. Both the Domain Controller and the Exchange Server were installed on two separate Win2k3 32bit (x86) operating systems according to Daniel Petri’s guide (Petri, 2008). The Relying Party, Secure Token Sever and Authorisation Server were installed on a Microsoft Windows Server 2008 (Win2k8) x86.

Applications
The Server Applications were written in Visual Studio 2008 using C# to access the functions in the .Net Framework.

Code Snippet 1: Setting the certificates in the STS

```
public class MySecurityTokenService : SecurityTokenService
{
    const string SigningCertificateName = "CN=sts2008.netbox.honours.net";
    const string EncryptingCertificateName = "CN=rp2008.netbox.honours.net";
    const string AddressExpected = "https://corporateactionman.homeip.net/MultiAuthRP";
}
```

Code Snippet 2: Setting the STS certificate in the Web.Config file

```
<microsoft.identityModel>
    <issuerNameRegistry type="Microsoft.IdentityModel.Tokens.SimpleIssuerNameRegistry,
        Microsoft.IdentityModel, Version=0.5.1.0, Culture=neutral,
        PublicKeyToken=31bf3856ad364e35" />
    <serviceCertificate>
        <certificateReference findValue="CN=sts2008.netbox.honours.net",
            storeLocation="LocalMachine" storeName="My"/>
    </serviceCertificate>
    <audienceUris>
        <add value="https://corporateactionman.homeip.net:4433/MultiAuthSTS_Card/login.aspx"/>
    </audienceUris>
    <federatedAuthentication enabled="true"/>
</microsoft.identityModel>
```
The Secure Token Server was installed by downloading the Geneva Framework (Microsoft Connect, 2008) which requires Win2k8, IIS 7.0, .Net Framework 3.5 and Visual Studio (2005 or 2008). The “Web Application with Multiple Signin Methods” sample was installed from the “End-to-end Scenario” samples list. IIS 7.0 was then configured with non-standard ports of 4433 and 8080 allocated to the SSL and HTTP services respectively to ensure that port-forwarding could take place behind the Router’s Network Address Translation (NAT). Digital Certificates were created according to John Howard’s article (Howard, 2005) for the STS and the Relying Party and installed.

Using Visual Studio 2008 the “STSCard” visual studio project: endpoints and certificates were then altered in the web configuration file (Code Snippet 2) and Security Token Service Class (Code Snippet 1) in line with Vittorio’s blog (Vittorio, 2008) and the MSDN documentation.

The final step was to expose the Authorisation Service to the STS client using Windows Communication Foundation (WCF), as the Geneva Framework relies on .Net 2.0 for its configuration, an Adapter Pattern (Barclay & Savage, 2004) was written in another project file to expose the .Net 3.5 WCF service as an internal reference.

Authorisation Service was written on a Win2K3 Development Server and published on to a Win2k8 production server using Visual Studio 2008. As discussed section 4.4.1 the key component of the Authorisation Service is an Attempt Manager which is written using the WCF application template. The Attempt Manager (Figure 13: Authorisation Service Class Diagram in Appendix B) has 3 main components:

- **An asynchronous event handler** that is written using the inbuilt asynchronous functions which come with WCF according to the design in the Appendix B sequence diagram. The handler also uses the Asynchronous Operation Library written by Dan Rigsby (2008); this library allows unique attempts to have a limited life-span and current status using the PendingOperationController().

- **The Attempt class** (Figure 13) described in the Authorisation Service class diagram is responsible for two aspects of the out-of-band loop, persistence and end-to-end encryption (Buchanan, Layered encryption, 2007). Persisting XML Attempts takes place using .Net’s inbuilt XML Serialisation (Microsoft, 2008) according to the design choice in section 4.3.2. Reflection (Microsoft, 2008) gives responsibility to the Attempt Class for all encryption and decryption of Attempt information that will be sent Over-The-Air (OTA) to the Mobile Device as described in the Code Snippet 3.
• **WAP Push** is handled by an adapter class which references a WAP Push Library that was modified from the code provided by Adam Bird (Bird, 2004). The WAP Push Library accesses the Bird’s Esendex SMSC API through a SOAP web service. The Attempt manager sends a URL (discussed further on) to the Mobile device which is due and previously highlighted in section 4.3.2 to the restricted number of characters available in Esendex’s Push Message Service.

The attempt data is exposed via a WCF RESTful (Bagby, 2008) service. The mobile device locates the attempt data using the URL it received from the WAP Push (or straight SMS) message it received from the Attempt Manager. Attempt data is the persisted encrypted xml created by the Attempt class.

The final part of the Authorisation Service is an ASP.Net Web Service which acts as an adapter to the Attempt Manager’s WCF Service as described in the Design Tradeoff’s section.

• **The Relying Party**, very much like the STS, is installed using the Geneva Framework. It uses the standard SSL and HTTP ports. Visual Studio 2008 was used to alter the login form and configuration file in the “**Relying Party**” visual studio project. The endpoints and Digital Certificates of itself and the “**Trusted Issuer**” were modified accordingly.

---

### Code Snippet 3: Encrypting an Attempt object using Reflection

```csharp
private PropertyInfo[] properties;
...
this.properties = this.GetType().GetProperties();
...

key = Convert.FromBase64String(this.serverSecretKey);
cipher = new PaddedBufferedBlockCipher(new CbcBlockCipher(new AesLightEngine()));
cipher.Init(true, new KeyParameter(key));
foreach (PropertyInfo property in properties)
{
    if (property.Name.Equals("UserName") || property.Name.Equals("SocialSecNum")
    || property.Name.Equals("PhoneNumber")
    || property.Name.Equals("RelyingParty"))
    {
        byte[] inBlock = System.Text.Encoding.UTF8.GetBytes((property.GetValue(this, null)).ToString());
        byte[] outBlock = new byte[cipher.GetOutputSize(inBlock.Length)];
        try
        {
            int outLength = cipher.ProcessBytes(inBlock, 0, inBlock.Length, outBlock, 0);
cipher.DoFinal(outBlock, outLength);
        }
        property.SetValue(this, Convert.ToBase64String(outBlock), null);
    }
}
```
5.3 Mobile Clients

This section discusses the two technologies used in the implementation of the project and the challenges encountered. The two platforms of JavaMe and the .Net Compact Framework are discussed in this thesis and implemented (only partially in one case) in the project. Each platform is competing with the other in a sense; it is therefore important to develop the project with a view to see that the infrastructure can cope with different client-types and to see how robust each client platform might be when used in as an authentication factor in a Federated Identity metasystem.

5.3.1 Methodology

To ensure interoperability between the clients and the infrastructure, the implementation was broken into phases. The first phase was to establish communication between a .Net web service and JavaMe Client; Phase two involved the implementation of a non encrypted end-to-end out-of-band loop between the Authorisation Service and the JavaMe Client; The third iteration was a repeat of phase two with the Windows Mobile Client; Phase four was the encryption and decryption of the XML data fetched by the Windows Mobile client; The fifth stage was a mirror of stage four with the JavaMe client and finally the sixth phase was the incorporation of the out-of-band loop into Federated Identity Model (see section 5.1.2). As JavaMe was not implemented in the fifth stage due to problems encountered with BouncyCastle API interoperability, the focus for the rest of the implementation will be on the .Net Compact Framework implementation.

Code Snippet 4: Activating the application with an SMS message

```csharp
const string _appId = @"corporateaction.homeip.net.authenticator.receive-sms";

private MessageInterceptor _sMSInterceptor;

if (MessageInterceptor.IsApplicationLauncherEnabled(_appId))
{
    _sMSInterceptor = new MessageInterceptor(_appId);
}
else
{
    _sMSInterceptor = new MessageInterceptor(InterceptionAction.NotifyAndDelete);
}
_sMSInterceptor.MessageReceived += new MessageInterceptorEventHandler(SMSInterceptor_MessageReceived);
```
5.3.2 Operation

It should be noted that in this project the .Net Compact framework handles SMS activation of an application differently to JavaMe see section 3.7.2 and section 3.8.2. To implement automatic activation of an application the code snippet (code snippet x) was used:

It can be seen from (Figure 15: Mobile Client in Appendix B) that the Mobile client uses the same Attempt Class that is present in the Authorisation Service and the Attempt Retrieval Service. When the application runs it uses the SMS body to fetch the XML from the RESTful web service as a data stream which it then deserialises as an Attempt object, which can be observed in Code Snippet 5. Once an attempt object is in memory, the attempt information can be decrypted for the user to make an authorisation decision.

Code Snippet 5: Mobile Fetching encrypting attempt data and deserialising

```csharp
void SMSInterceptor_MessageReceived(object sender, MessageInterceptorEventArgs e)
{
    SmsMessage smsMsg = e.Message as SmsMessage;
    Uri xmlUri = new Uri(smsMsg.Body);
    WebRequest webReq = HttpWebRequest.Create(xmlUri);
   WebResponse webResp = webReq.GetResponse();
    Stream data = webResp.GetResponseStream();
    using (StreamReader reader = new StreamReader(data))
    {
        // Go and get the XML from the RESTFUL Server
        string xml = reader.ReadToEnd();
        XmlRootAttribute xRoot = new XmlRootAttribute("Attempt");
        xRoot.Namespace = "http://schemas.datacontract.org/2004/07/AttRetrievalService_2";
        xRoot.IsNullable = true;
        XmlSerializer ser = new XmlSerializer(typeof(Assert), xRoot);
        Attempt att = (Attempt)ser.Deserialize(reader);
        // Do something with the Attempt data...
    }
}
```

5.4 Conclusion

This system visibly suffers from basing all of the services on one virtual host machine with one Network Interface Card (NIC); it is also burdened as it runs behind Network Address Translation (NAT) and uses DynDNS registered addressing to communicate between the virtual servers and services. Evaluation of network latencies and server loads should take place on this prototype and is discussed in section 6.3. Other weaknesses include the system’s reliance on the SMS Centre to activate the application on the phone and need for the device to fetch authentication attempt information from a server. Encryption of the response from the device to the Authorisation Service was also not implemented given time constraints which is a major weakness.
6 Evaluation

6.1 Introduction
This Chapter is divided into two parts; the first looks at the effectiveness of the implementation in regard to the Key aim, objectives and goal set out in section 1.2; the second describes and justifies a testing methodology for measuring the implementation in terms of integrity, availability and performance.

6.2 Effectiveness of Deliverables

6.2.1 Key Aim
The main purpose of this thesis and project was the proposal of an out-of-band self-authentication approach to online identity proof which addresses the matters highlighted in section 1.1.

The key aim of this thesis has been met in-part; it has described an out-of-band self authentication system which does improve the information security cornerstones of Confidentiality through Accountability in an InfoCard system. It is however fraught with problems: the non availability of connection to the Out-of-Band device (B'Far, 2005); the added complexity of Security in a wireless communication channel; and that the elevation of personal devices to mobile computers has increased the opportunity for Malware writers (GTISC, 2008)

6.2.2 Objectives
The project implementation had two objectives that are listed as follows: Implement a Proof of Concept prototype similar to the system proposed by Wu et al (2003) using the more up-to-date paradigms of WAP Push and Web Services; Specify and build clients in JavaMe and Windows Mobile 6 that will use that prototype infrastructure to complete the out-of-band authentication loop.

The Proof of Concept outlined by Wu et al (Wu, Garfinkel, & Miller, 2003) differs quite markedly to the Federated Identity Metasystem that was implemented for this Project in that the Identity Provider proposed by Wu et al actually acts as a Web Proxy for the Relying Party and no exchange of Tokens actually takes place. For interoperability-sake the choice of Web Service
protocol was constrained by the JavaMe Client to the Basic Profile 1.1 whereas the Windows Mobile .Net CF client is able to handle WS Security in the form of WSHttpBinding. Another interopability challenge highlighted in section 4.3.2 was encountered when trying to access the WCF published Web Services Description Language (WSDL) document, as WCF compiles the WSDL in a modular manner consisting of several WSDLs the JavaMe NetBeans development platform is unable to cope with multiple WSDLs. A work around to dynamically flatten the modular WSDLs into one document is described by a work-around of flattening the WSDL in Chiesa’s article (2008) however it seemed too complex for this project. The work around implemented and described in the Design tradeoffs section is An ASMX .Net 2.0 web service that is based solely on the WS-I Basic profile which acts as an intermediary between the WCF service and the client. This option proved to be serendipitous when integrating with the Federated Identity System.

Encryption Problems with JavaMe and Bouncy Castle prevented the final implementation of JavaMe part of the 1st objective and the failure to produce the one of the deliverables described in the project proposal

It was initially envisaged that the Identity Provider had access to an Identity Store like a Microsoft Exchange Server X500 Certs (Vittorio, 2008), unfortunately time did not allow for this.

6.2.3 Usability testing Secondary Goal

To gauge user feelings toward a Federated Identity Metasystem with an out-of-band loop, the Project Proposal outlines a desire to conduct a brief usability test of Managed Cards in this Thesis’ context. This goal was partially successful as a Usability test was conducted on a small group of nine non-heuristic users, using Personal Cards in a Managed Card Context described in subsection 6.2.5.

Usability test Methodology

In testing users perceptions of Federated Identity Proof it was hypothesised that the test should be made to look as real as possible as a user’s perception of online security and trust might have a significant impact on the results of the usability test. Consider this trust-perception result link it was decided to wrap both of the Geneva framework’s sample STS and Relying party’s ASPX pages in HTML which give them the appearance of commercial sites; in the case of the Relying Party, it was made to look like an online bank with HTML structure taken from an actual bank; the STS was made to look as trustworthy as possible using dark blue colours and the Government Building icon from Microsoft Visio to present a feeling of trust through symbolism. Users were subjected to a brief questionnaire prior to taking part in the story-boards. More
information on this testing is available in Appendix A. All 9 users took part in the following three scenarios:

- Log on using username and password and no out-of-band loop
- Log on using Personal Information Card and no out-of-band loop
- Log on using a Personal Card in a Managed Card context and no out-of-band loop
- Log on using a ‘Managed Card’ (as above) with an out-of-band loop

Users were then subjected to a brief questionnaire about their perceptions of online trust and if any of their perceptions had changed.

**Results**

7 out of 9 of the users in this small study (77%) transacted cash over the internet at least once a week; 7 out of 9 of the users felt a low trust when carrying out confidential transactions over the internet; Only 2 of the group were concerned about the security of online transactions; The results were totally mixed with a regard to change in attitude to online security in the past 5 years; All the surveyed group seemed concerned about Identity theft; All the group placed the phone authentication technique as having the highest trust rating, most of the users thought that password authentication was more trustworthy than managed cards, managed cards came third and personal cards came fourth; All the group thought that the Identity Provider had a high importance; All the group thought that their trust levels would increase if any of the techniques were implemented; The user-friendliness was generally positive in the study but one user thought the network latency of the out-of-band loop was potentially troublesome.

**Analysis**

It is interesting that all of the users in the study felt most comfortable with self-authentication and that personal and managed cards all came after password authentication in the user groups perception of trust. It also seems that users see that the Identity Provider is highly important without knowing the mechanisms behind it, which suggests that a sound mental-model (Benyon, Turner, & Turner, 2005) could be formed without the need for extensive training. All These results definitely point to a need for further research in a larger study.

### 6.3 Testing

This project evaluation is focused on the performance testing on the Mobile client device. It was initially envisaged that several categories of measurement would be conducted on this Projects Federated Identity Structure and its mobile client however in the time given it was decided to
settle on one type of test in order to concentrate on the quality of the results, more discussion on this decision is undertaken in Chapter 7.

6.3.1 Justification
Battery Life is part of the Mobile Development Landscape and encryption integral part of Confidentiality and Integrity of over-the-air communications. Public Private Key encryption and decryption has always been viewed as Resource Intensive on the CPU and the Memory, even more of an issue in a Mobile Environment. Not enough testing has been done on Mobile applications in this area (B'Far) it therefore seems appropriate to conduct this sort of measurement.

6.3.2 Methodology
A decision was taken to create an application based on the original Mobile Client Application That would deserialise a locally stored XML document to simulate the RESTful fetch of the encrypted Authorisation Attempt Data – this application was designed to deserialise with and without encryption by user selection at run-time. A cursory evaluation was conducted on the Emulator versus the actual device and the impact on both appeared to be very similar.

An application acbTaskman was available to purchase which logs performance statistics per mobile application into a CSV file. The CSV results were then imported into a spreadsheet to display the results in line graphs of percentage use which follows in the next section.

It was decided that the program execute 10,000 iterations to try and run the battery down.
6.3.3 Results
Both charts below show the percentage of total processor and memory usage over time.

**Processor and Memory usage without encryption**

- % Memory
- Process CPU (%)

**Processor and Memory usage with encryption**

- % Memory used
- Process CPU (%)

![Graph showing processor and memory usage](image-url)
6.3.4 Analysis
The troughs show sleep mode has been initiated on the device, which is common to all Windows Mobile Pocket PC based devices. Ignoring these troughs it is evident that the program running without encryption uses more memory and slightly less CPU time over the period of the testing and that the encryption enabled program uses slightly less memory and slightly more CPU over the 10,000 iteration period. What is more evident though is the time taken to run each program; the non encrypted program took only 4 minutes and 30 seconds complete, whilst the encrypted program took hours and even ran the battery down.

6.3.5 Conclusions
Although it is not ideal to run logging application and test application on same device, time did not allow a more exact solution. It would also have been better to rework the testing application to ensure that the device does not time-out as the operating system intends it should. It is clear from the analysis that using public and private key encryption and hashing has a considerable effect on the performance and in this context users who undertake a great deal of self-authentication will have a shorter battery-life experience from their device. It is interesting to note that the test application developed is not specific to the Project Context but is specific to BouncyCastle serialised Public and Private Keys – could therefore be used to test other Windows Mobile devices.
7 Conclusions

7.1 Summary
This thesis has explored the topic of identity proof across the internet; it has used a project which implemented a Federated Identity metasystem with an out-of-band authorisation factor. BouncyCastle decryption performance was measured on a Windows Mobile device versus the same algorithm that did not contain decryption and the results were analysed with conclusions that follow in section 7.3. A cursory usability study was conducted to highlight further areas of research for Windows CardSpace and the Federated Identity metastructure as a concept.

7.2 Critical Analysis of Technologies Used
The original project proposal outlined a desire to compare the Windows Mobile platform against a JavaMe platform when used in an out-of-band authorisation loop, with an inferred focus on Decryption performance. The implementation of the project failed to get the JavaMe platform to interoperate with the BouncyCastle API because of a conflict in the java.math package, a few articles pointed to a successful implementation of the BouncyCastle lightweight API but time did not allow a solution to the problem encountered.

7.2.1 Ease of Development
During implementation it was found that developing JavaMe applications for Sony Ericsson devices was generally not as easy as developing .Net Compact Framework applications for the Windows Mobile Device. The deployment application for the Sony Ericsson device did not work as highlighted in section 3.7.2, in contrast all of the Development tools for the Windows Mobile 6 SDK functioned as expected and in addition the Windows Mobile 6 SDK had a cellular emulator which allowed simulated OTA SMS messages to be sent to the phone, and a simulated 3G connection from the phone to the internet via the development machine’s COM port. The user-interface development was marginally easier using the NetBeans Visual Midlet Designer.
7.3 Analysis of Results
It is clear from the usability study that users feel more comfortable with self-authentication as part of a Federated Identity metasystem than either managed cards or personal cards on their own and that there was some confusion as to the relevance of the cards over usernames and passwords. Some degree of learning might be necessary for users to adopt the cards fully. The performance impact of decryption on the mobile device is quite apparent which indicates an area for further study.

7.4 Future Work
Given the time constraints versus the ambitiousness of this project, there is great deal of future work that might be conducted. More time could be spent on finding a solution to the JavaMe and BouncyCastle interoperability problems with a view to comparing the Decryption performances of both platforms. The performance study yielded a result which suggests that a tool similar to the Microsoft Hopper tool would be needed to thoroughly test the system for inadvertent user interaction.

At the start of this project it was envisaged that measurements like Network latency and Server resilience would be taken and analysed, it would be interesting to see the impact of the Federated Identity System and the impact of the out-of-band loop on overall response times perceived by the user.
8 Bibliography


Hussain, A. (2006, November 12). *NatWest shuns ID fraud victims: With identity theft on the increase, the big high street lenders are becoming less willing to accept responsibility*. Retrieved November 11, 2008, from TIMES Online: http://www.timesonline.co.uk/tol/money/consumer_affairs/article633729.ece


Appendix 1: Project Overview

Initial Project Overview
Title: Comparing and Contrasting different mobile phone technologies when implementing Out-Of-Band Authentication to a Web Portal using social security numbers to identify users.

Overview of Project Content and Milestones
To research main technologies and motivations involved in implementing out-of-band authentication with a Mobile Phone.

Implement and test network infrastructure that will support the development of an out-of-band authentication system with a Mobile Phone.

To develop a Java ME (formerly J2ME™) and a Windows Mobile® 6 phone application that will both act as clients in the network infrastructure implemented above.

Conduct analysis of the network infrastructure and the clients to determine areas where measurements can be successfully and relevantly applied.

Undertake research relating to the measurement areas identified above to reveal analysis tools and methods.

Implement analysis tools and methods revealed above.

Conduct initial tests to determine the effectiveness of the study.

Undertake a full set of tests that will be used to compare the two mobile phone client applications.

Compare findings of tests with existing literature.

Reflect and draw conclusions from the overall project.
The Main Deliverable(s)
An implementation of a distributed system is proposed that demonstrates out-of-band authentication using a mobile phone. An arbitrary web portal will request the authorisation of a login to a remote service which verifies if a user has authorised the login via a mobile phone. It is also proposed to write a paper that details the qualitative and quantitative comparison between Java ME and Windows Mobile® 6 when used in an out-of-band authentication system.

The Target Audience for the Deliverable(s)
The target audience for the implementation will be non-heuristic users who have identified themselves as being unconfident in the security of online transactions. The quantitative and qualitative analysis between the two types of Mobile phone platform is targeted at developers who wish to implement authentication using a Mobile phone and to researchers in the field of Information Security and Network Security.

The Work to be Undertaken
It is proposed to investigate, specify, design and implement of a Secure Authentication Server. This authentication server will allows authentication to an arbitrary Web Portal using one of two mobile phone client applications. The proposal is then to investigate, specify, design, implement, test, analyse and evaluate two mobile phone client applications developed with the competing Java ME and Windows Mobile® 6 technologies.

Additional Information / Knowledge Required
To acquire knowledge of:

- Authentication Sever systems.
- Implementation of a distributed system using a European Mobile phone network.
- Measurement tools and methods in a distributed computing environment.
- Mobile phone platform development practices and methods.
- Current user attitudes towards online transactions and mobile distributed applications.

Information Sources that Provide a Context for the Project
At the BCS Symposium on Intelligence in Security and Forensic Computing held at Napier University in 2006. Dr Geraint Price gave a presentation called “Security without Authentication?” At the symposium, Dr Price proposed that it might be almost impossible to
verify the identity of a remote user as all data is digitised and can be replicated. This project aims to use current research and practices to implement a “judicious use of authentication” highlighted by Dr Price. By tying the geographic location of the user to an authentication request via a mobile phone using a system proposed (and not implemented) in “Secure Web Authentication with Mobile Phones” by Wu, Garfinkel and Miller (2003) it is anticipated that a solution like this (or similar) will help in the on-going challenge to verify user identity across a distributed network. A similar and implemented piece of research called “Using a Personal Device to Strengthen Password Authentication from an Untrusted Computer” by Mannan and van Oorshot (2007) [3] tries to address this issue with Bluetooth through the authenticating terminal, whilst this manner of user verification is interesting and could be used to bolster the solution provided by Wu, Garfinkel and Miller (2003) [2] it will not be considered as a basis for this project.

The Importance of the Project

“IDENTITY THEFT topped the list of consumer complaints about fraud, according to the U.S. Federal Trade Commission’s annual report for 2005, accounting for 255,000 of the more than 686,000 complaints filed with the agency in 2005” (Tamara Dinev, 2006, p.2) [1].

This project is significant to Network Security community by presenting qualitative and quantitative evaluation of Java ME in comparison to Windows Mobile® 6 when implementing an out-of-band authentication system.

The Key Challenge(s) to be Overcome

Implementation and testing of network authentication infrastructure prior to development of client applications.

Development and testing of Mobile phone client applications and their interactions with the implemented network infrastructure.

References

Appendix 2: Initial Project Monitoring Form and Project Diary Sheets
Appendix A: Usability Testing

Background
You are the online customer of *Conventional Fife (CF)* a well respected and regarded Bank with Headquarters in Perth. Conventional Fife guards its reputation for respecting Customers Confidentiality, the Integrity of their clients’ Bank Accounts and for providing their clientele with extremely high levels of available access to Conventional Fife’s services. Given CF’s attitude to information security, they are currently trialling different ways of online authentication and have identified you as a loyal customer who is willing to help them through this trial.

Scenarios
You have volunteered your services to participate in the following scenarios in return for a Lovely Sunday Walk in Mull on the 14th of December. Prior to starting any of the Scenarios you will be asked 5 questions, on completing each Scenario you will be asked for any thoughts and on completing all 4 Scenarios you will be asked a further 5 questions.

Scenario Consideration - Important
On accessing these Web Sites you will be presented with a Browser Warning Screen regarding Security – this is normal in a Trial Situation and you can continue on to the Web Site by clicking the link next to the red shield and not the green shield – you must try your hardest not to let this affect your perceptions of trust and security. Note that in a non trial environment, you are encouraged if not urged to read these warning messages carefully and take the appropriate action (most likely clicking the link next to the green shield).
Log on using username and password and no out-of-band loop
1. Open a browser and browse to http://corporateactionman.homeip.net/MultiAuthRP.

2. You are presented with Conventional Fife’s Online Login Page. Proceed by Clicking the Blue Log in button.

3. Click the bottom link next to the red shield labelled “Continue to this website (not recommended).”

4. You are presented with the Log In Form, enter the User name JohnDoe and the password JohnDoe123.

5. Click the Button Labelled Log In.

6. Click the bottom link next to the red shield labelled “Continue to this website (not recommended).”

7. You will see a table that represents a successfully Logged on User.

8. Please close the browser entirely.

Log on using Personal Information Card and no out-of-band loop
1. Open a browser and browse to http://corporateactionman.homeip.net/MultiAuthRP.

2. You are presented with Conventional Fife’s Online Login Page. Proceed by Clicking the Grey CardSpace Authentication Arrow Logo.

3. Click the bottom link next to the red shield labelled “Continue to this website (not recommended).”

4. You are presented with the Log In Page, Click the purple InformationCard i Logo.

5. Windows CardSpace opens – before doing anything look at the computer screen and try to navigate to a different window whilst CardSpace is open, also listen to the Computer to see if you hear anything different.

6. Click on the Grey Icon with a yellow star in the top left called Add a card.

7. Select “Create a Personal Card”.

8. Enter a name for the card, then enter your First Name (highlighted in red).
9. You will return to the original screen. Make sure the card you created is highlighted in Blue and click Send

10. You will return to the Identity Provider Site Please Wait.

11. Click the bottom link next to the red shield labelled “Continue to this website (not recommended).”

12. You will see a table that represents a successfully Logged on User

13. Please close the browser entirely.

**Log on using a Personal Card in a Managed Card context and no out-of-band loop**

In this Scenario you will get a managed card from your Trusted Identity Provider – in this case your Trusted Identity Provider is Napier University.

1. Report to your Scenario supervisor who is a representative of Napier University, ask for them to show you some Photo Identification, and request a ‘managed card’.

2. The IP representative will ask you for Photo Identification and upon presenting, this id, they will provide you with a Managed Card. Feel free to ask any questions about this process as they occur to you.

3. On receipt of this managed card, Load this card by going into:

   a. Control Panel -> Windows CardSpace -> Restore Card -> Browse to the Following Location C:\\<Your Name>\ and select the file with the extension .crd

4. Open a browser and browse to [http://corporateactionman.homeip.net/MultiAuthRP](http://corporateactionman.homeip.net/MultiAuthRP) .

5. You are presented with Conventional Fife’s Online Login Page. Proceed by Clicking the Grey CardSpace Authentication Arrow Logo.

6. Click the bottom link next to the red shield labelled “Continue to this website (not recommended).”

7. You are presented with the Log In Page, Click the purple InformationCard i Logo.
8. Windows CardSpace opens – before doing anything look at the computer screen and try to navigate to a different window whilst CardSpace is open, also listen to the Computer to see if you hear anything different.

9. Click on the Grey Icon with a yellow star in the top left called Add a card.

10. Select “Create a Personal Card”.

11. Enter a name for the card, then enter your First Name (highlighted in red).

12. You will return to the original screen. Make sure the card you created is highlighted in Blue and click Send.

13. You will return to the Identity Provider Site Please Wait.

14. Click the bottom link next to the red shield labelled “Continue to this website (not recommended).”

15. You will see a table that represents a successfully Logged on User.

16. Please close the browser entirely.

Log on using a ‘Managed Card’ (as above) with an out-of-band loop

Same as above but the Supervisor will give you access to a mobile device – after you have sent your card using cardspace a message will appear on the phone asking you to verify the request.
Questionnaire

1. How often do you conduct business on the internet involving a cash transaction?
   a. Every Day?
   b. Twice a Week?
   c. Once a Week?
   d. Twice a Month?
   e. Once a Month?
   f. Once every 3 Months?
   g. Never

2. On a Scale from 1 to 5 how would you rate your trust-level of confidential transactions carried out over the internet 1 being very trustworthy and 5 being very untrustworthy?

3. On a Scale of 1 to 5 how concerned are you about the security of online transactions?

4. How has your attitude to online transaction security changed over the past 5 years?
   a. Increased Markedly
   b. Increased
   c. Stayed the same
   d. Decreased
   e. Decreased Markedly
5. How would you rate your concern for Identity theft?
   a. Extremely concerned
   b. Concerned
   c. Not bothered
   d. Haven’t given it any thought.

6. Place the authentication techniques shown in the trial into a sliding scale of trustworthiness.

7. In this scheme how would you rate the importance of an Identity Provider?
   a. High
   b. Medium
   c. Low

8. If any of the systems shown today were implemented, rate the change in your trust levels.
   Improve slightly

9. In terms of usability how would you rate each of the techniques?
   a. Highly user-friendly
   b. User-friendly
   c. Not user friendly
   d. Extremely user-unfriendly

10. Any other Comments?
Appendix B UML Diagrams

Figure 11: Metasystem Sequence Diagram
Figure 13: Authorisation Service Class Diagram

WCF SOAP AuthorisationService
(WS-* Profile)

Figure 12: Intermediary Authorisation Service

ASMX SOAP Phone Authorisation Service WS-I Basic Profile
Figure 14: Restful Attempt Retrieval Service

RESTful Attempt Retrieval Service

Figure 15: Mobile Client
Appendix C: Mobile Topologies

Figure 16: WAP Push topology
Figure 17: WAP 1.0 Protocol Stack (Orange Partner)
Figure 19: WAP 2.0 Protocol Stack (Orange Partner)

Figure 18: i-Mode Protocol Stack
Appendix D: Mobile Technologies

Figure 20: Windows Mobile 6 SDK Documentation (Microsoft, 2008)
Appendix E: Web Service protocols

Web Service Protocols
This thesis discusses 3 types of Web Service Protocols, two based on Simple Object Access Protocol (SOAP) and one based purely on resources (for example XML) located and described by their URIs called Representational State Transfer (REST).

The two SOAP protocols considered in this project are the Basic Profile Version 1.1 published by Web Services Interoperability Organisation (WS-I) and the Web Services Security Profile discussed in section 3.5.4.

WS-Policy version 1.2
WS-Policy is a recommendation by W3C written by BEA Systems, IBM, Microsoft, SAP and VeriSign (2006). WS-Policy defines a framework that uses XML to describe the policy of a Web Service, it comprises of the following three components:

- **A Policy Assertion** according to the framework identifies a behaviour requirement of a policy subject. Nolan describes it (2004) as the basic unit of policy and that it could be thought of as an instruction for processing, the example he gives is the assertion that the message be encrypted.

- **A Policy Alternative** is described by the framework as “a logical construct which represents a potentially empty collection of policy assertions”. Nolan (2004) provides a clearer view of an acceptable combination of assertions that form a complete set of instructions for processing.

- **A Policy expression** is the root element of the document that encompasses a set of Policy Alternatives and possible nested Policy expressions. The policy can be associated with a namespace.

WS-Metadata Exchange
The WS-Metadata Exchange is a specification developed by BEA Systems, IBM, Microsoft and SAP (BEA Systems, IBM, Microsoft, SAP, 2004) that describes how endpoints can exchange specific metadata allowing them to interact. In the context of this Thesis it defines SOAP messages that can retrieve WS-Policy information.
WS-Security (WSS)
WS-Security originally developed by Microsoft, IBM and VeriSign is a standard now published by the Organization for the Advancement of Structured Information Standards (OASIS) that encompasses the following specification and profiles (OASIS, 2006):

- WS-Security Core Specification 1.1
- Username Token Profile 1.1
- X.509 Token Profile 1.1
- SAML Token Profile 1.1
- Kerberos Token Profile 1.1
- Rights Expression Language (REL) Token Profile 1.1
- SOAP with Attachments (SWA) Profile 1.1

WS-Trust
The WS-Trust specification developed by IBM, Microsoft and Actional, BEA, Computer Associates, Layer 7, Oblix, OpenNetwork, Ping Identity, Reactivity, and VeriSign (Actional, BEA Systems, Computer Associates, IBM, Layer 7, Microsoft, Oblix, Ping Identity, Reactivity and Verisign, 2005) defines extensions to WS-Security for providing a framework that can allow requests and issuances of security tokens to facilitate the trust relationships like those already described between the Relying Party, the User and the Identity Provider.