VTT RESEARCH NOTES 2596

Aki-Petteri Leinonen

Identity management for web-enabled smart card platform

VTT TIEDOTTEITA – RESEARCH NOTES 2596

Identity management for web-enabled smart card platform

Aki-Petteri Leinonen

ISBN 978-951-38-7780-4 (soft back ed.)
ISSN 1235-0605 (soft back ed.)

ISBN 978-951-38-7781-1 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1455-0865 (URL: http://www.vtt.fi/publications/index.jsp)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 5, PL 1000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 5, PB 1000, 02044 VTT
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O. Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 4374

Technical editing Marika Leppilahti

Text formatting Raija Sahlstedt

Kopijyvä Oy, Kuopio 2011

Aki-Petteri Leinonen. Identity management for web-enabled smart card platform. 2011. VTT Tiedotteita – Research Notes 2596. 64 p. + app. 2 p.

Keywords identity management, smart card, mobile web services, single sign-on

Abstract

The amount of sensitive information stored in different online services is rapidly growing in traditional web applications and also in mobile services. Most of these services control their own authentication credentials, increasing the number of credentials that the user needs to manage. Previous literature shows that when the amount of passwords grows, users tend to create weaker passwords or reuse passwords for different services. This exposes users to security threats. Attacks target weak passwords and compromised security might result in a domino effect, with one exposed password giving access to multiple services. In addition to usability issues, the mobile platform is vulnerable to physical attacks if the device is lost or stolen. This creates a need for a secure credential management platform for mobile devices that addresses these problems and creates a usable environment for the management of credentials. One such solution could be provided by a secure element inside the mobile device. The secure element is special hardware that provides secure code execution for the mobile platform, as in existing smart card platforms.

This study is based on work carried out at VTT Technical Research Centre of Finland, in which a prototype version of a single sign-on Java Card application was created with a Java Card 3 generation emulator. An earlier prototype gave perspective for this work and served as an additional evaluation point for the constructed prototype. The purpose of this study is to find out whether a secure element with support for web-connected services can be used to provide a user-centric credential management platform in a mobile phone. This main question can be divided into three questions that need to be answered: What are the users' password management strategies, what requirements can be identified for a user-centric credential manager inside a secure element and can this solution be implemented with existing technology?

In order to find out password management strategies and existing implementations, an extensive literature review is conducted. With respect to the use of password management strategies, the literature review indicated that users circumvent security methods because they consider that these methods take too much of their time and other resources compared to the perceived gains. It was also revealed that the authentication procedure must follow the user's mental model and not restrict the primary task the user needs to achieve. The requirements for the credential model are identified from the literature review and a single sign-on protocol is chosen to be the approach in this work. A prototype application that allows users to authenticate to different services with a single sign-on, and which also demonstrates two-factor authentication, is built and evaluated along with the earlier prototype. The hardware platform used in the prototype is a secure microSD memory card with an embedded Java Card 2 smart card chip. The prototype application built in this work shows that the credential manager application can be implemented in an open manner with a single sign-on protocol. The prototype shows promising results and offers a solid platform for identity management in a mobile phone when the number of services increases. That said, the need for further research on credential storage use in client applications is also identified.

Preface

This study was written at VTT Technical Research Centre of Finland during 2010 and spring 2011 as a master's thesis. The work was done as part of the Smart Urban Spaces project, which is an ITEA2-funded European-wide effort to introduce interoperable e-city services.

Part of this study is based on work done in the Worldcom project, which was an internally funded VTT project regarding the creation of a prototype application on the Java Card 3 platform. I express my gratitude to Dr. Marko Jurvansuu for bringing me on board the project, and to the other members of the project team: Kai-Wen Chan, Ville Köykkä, Jari Ruokonen and Miikka Saukko. The Worldcom project provided perspective for this work with a proof-of-concept prototype. The Worldcom prototype was included in the evaluation along with the prototype constructed for this work.

I would like to thank my supervisor, Assistant Professor Jouni Markkula from the University of Oulu, for his guidance and encouragement during this work. I would also like to extend my thanks to my second reviewer, Professor Jouni Similä, also from the University of Oulu, for commenting on this thesis. My gratitude also goes to Dr. Tuomo Tuikka and Mr. Erkki Siira at VTT who have guided me through this work and provided valuable comments, and to all my colleagues at VTT who have helped me with this thesis.

Finally, I would like to acknowledge my friends and family for always supporting me in my work and studies. Especially, I would like to dedicate this thesis to the memory of my mother who passed away last spring: Thank you for always being there for me.

Oulu, 26 June 2011
Aki-Petteri Leinonen

Contents

Abstract 3

Preface 5

List of symbols 8

1. Introduction 9

2. Research problem and methods 11
   - 2.1 Research problem 11
   - 2.2 Research method 12

3. Background 14
   - 3.1 Usable security 14
   - 3.2 Browser-based threats 15
   - 3.3 Password phishing 17
   - 3.4 Identity and privacy management 17
   - 3.5 Secure element and trusted computing 18
   - 3.6 Smart card hardware 20
     - 3.6.1 Architecture 21
     - 3.6.2 Specifications 21
   - 3.7 Servlet environment 22
   - 3.8 Open credential protocols 22

4. Credential platform requirements 24
   - 4.1 Literature review 25
   - 4.2 Interoperability requirements 33
   - 4.3 Summary of the requirements 34

5. Prototype design 36
   - 5.1 Prototype definition 36
   - 5.2 Prototype platform 37
   - 5.3 Java Card 3 prototype 37
     - 5.3.1 Supported services 38
     - 5.3.2 Credential model 38
     - 5.3.3 Architecture 40
     - 5.3.4 Client application 42
   - 5.4 Java Card 2 prototype 42
     - 5.4.1 Services 43
     - 5.4.2 Credential model 45
     - 5.4.3 Architecture 46
     - 5.4.4 Client application 47

6. Evaluation 49

6.1 Evaluation criteria and design choices combined 49

6.2 Java Card 3 prototype 51

6.3 Java Card 2 prototype 54

7. Conclusion 57

References 59

Appendix A: 2D ticketing service

List of symbols

APDU	Application Protocol Data Unit
API	Application Programming Interface
ETSI	The European Telecommunication Standards Institute
GSM	Global System for Mobile Communications
HTTP/S	HyperText Transfer Protocol (Secure)
J2ME	Java Platform, Micro Edition
JSON	JavaScript Object Notation
OCF	The OpenCard Framework
PC/SC	Personal Computer/Smart Card
REST	Representational State Transfers
SATSA	Security and Trust Services API for J2ME
SE	Secure Element
SIO	Shared Interface Object
SIM	Subscriber Identification Module
TPM	Trusted Platform Module
VPN	Virtual Private Network
XML	Extensible Markup Language

1. Introduction

Most of us are using different kinds of online services containing different kinds of sensitive information about our personal life, ranging from online banking services to social media websites. All of these services are vulnerable to common security threats, such as information phishing and eavesdropping. Direct losses generated by phishing attacks amounted to over one billion US dollars in 2003 (Emigh, 2005). When the number of passwords to these services increases, users tend to reuse their passwords for different services and this can lead to a domino effect where one revealed password compromises several services (Gaw & Felten, 2006; Ives et al., 2004). If we focus on privacy, a different set of problems arises. Users send information to social services that are intended for different audiences. Consequently, users have different social contexts or roles in different situations, e.g. work role and private role. Users generally do not perceive social networking services as public places and paradoxically publish more information than they say they would be willing to reveal (Barnes, 2006). Users could therefore benefit from the privacy tools offered by these services; however, only a minority of users make effective use of them (Gross et al., 2005). Most importantly, users need an effective way to manage credentials to different services in order to reduce the number of passwords they need to remember and to avoid password reuse.

Java Card provides secure storage for credentials and is used widely in credit cards, access cards and other smart card applications demanding high security (Sun Microsystems, 2008). The hypothesis is that Java Card could provide secure storage for multiple credentials and thus serve as a platform for virtual single sign-on, as presented by Jøsang and Pope, giving the user more tools to manage passwords and privacy (Jøsang & Pope, 2005). In this work, a prototype application for multipurpose credential storage is constructed and evaluated. Prototype design and evaluation criteria are based on a systematic literature review conducted in the first phase.

The idea of an open multi-application secure smart card has been presented in earlier research. In April 2011, during the writing of this work, SIMAlliance released its specification for an Open Mobile API that is intended to provide a framework for services similar to the one presented in this work. SIMAlliance is a non-profit trade association focusing on secure mobile services, and it has supported the creation, deployment and management of secure mobile services (SIMAlliance, 2011).

This work is a continuation of work done by VTT Technical Research Centre of Finland in which a prototype version of a single sign-on Java Card application was created with a Java Card 3 generation emulator. The earlier prototype featured web-based single sign-on services including social media and e-mail services in addition to payment- and access-related mock-up services. It displayed promising results, but was implemented entirely with an emulator as a desktop application. This work continues the work done on the earlier prototype by identifying what is required from a mobile credential manager by performing a formal literature review and by implementing the application with SD-based Java Card technology in a mobile environment.

The study is structured as follows. In the first part, a literature review on password management strategies and credential managers is conducted. Information from the previous literature is used to establish the requirements for a credential manager application that provides a user-centric model for a secure element-based credential manager that can be used with web-based services. The built prototype is evaluated against the literature along with the earlier prototype to assess the technical feasibility of the system.

2. Research problem and methods

The growing number of web services means that users need to use more passwords. Users thus need an easy and secure way to manage them. The purpose of this study is to identify the requirements for a credential manager that focuses on identity management from a usable security viewpoint, and to test these requirements with a prototype credential manager. The target platform for the application is a mobile phone with a secure element, as mobile web services are seeing rapid growth and users usually always have their phone with them. Besides credentials, users also carry a significant amount of other personal data inside their mobile phones, which are vulnerable to theft and loss, and this makes mobile phones an ideal research target.

The prototype application built in this work is a mobile credential manager intended for web applications such as social media and e-mail reading, which secures the credentials with a secure computing platform. All of these services traditionally require different user credentials and provide no unified way to store credentials or implement authentication. The application implements single sign-on to different services using an open authentication protocol supported by these services. The prototype consists of two parts: a credential manager applet, which is installed on a smart card, and a client application that uses the credential manager applet transparently with a single login password. This prototype continues the work done with the earlier Java Card 3 prototype.

2.1 Research problem

The main problem area is identity management in a secure environment. In this context identity management is perceived as a broad issue, encompassing both the user and the service provider credentials and also information about the role for the user identity. This kind of information helps systems to decide for example which group the user is communicating with or what kinds of payment methods to use. The main research question is:

Can a secure element with web-connected services make identity and credential management easier and more secure for the user from the usable security viewpoint?

This main research question is divided into the following questions:

1. What are the users' identity management strategies and problems inherent in the existing credential managers?
2. What are the requirements for a secure element-based credential manager that addresses these issues?
3. Can the credential manager fulfilling these requirements be implemented with existing technology?

These questions identify the problem areas in credential management, password and privacy models and the problems related to the hardware platform and interoperability, all of which relate to the use of a physical device as authentication hardware.

2.2 Research method

The research method in this study is constructive and evaluative. According to Hevner et al. (2004), design science is by nature a rigorous search process where the designed artefact is derived from the available means to reach the desired ends. Also, Nunamaker et al. (1990) divide the research process into five separate stages. The first stage is to construct a conceptual framework that states a meaningful research question. After this the system architecture is developed, providing measurable requirements for the system evaluation. In the third stage, alternative solutions are analyzed and a system design is created, which is followed by the actual system building stage. The final stage is to observe and evaluate the system based on the conceptual framework. This study focuses on usable security, both in the literature review and prototype construction. (Kitchenham, 2004; Nunamaker et al., 1990).

In order to effectively use the available knowledge base and best practices for design, a broad perspective on both the currently available design approaches and the available background theories is needed. Therefore, a systematic literature review is conducted to define requirements for the construction of the prototype and its evaluation. The process described by Kitchenham (2004) is used to conduct the review. In the first step, a traditional literature search is carried out to gather background information. The background information is used to identify keywords that are then used to create database queries. The three most central databases are selected for the queries. First, the title and abstract of the discovered articles are analyzed. After that, the introduction and conclusion of the remaining articles are analyzed further to select the final set of articles for closer review.

Simon describes the design process as being a cyclic in nature, consisting of a number of build-and-test iterations before the completion of the final design artefact. This iteration cycle is shown in Figure 1 (Simon, 1996).

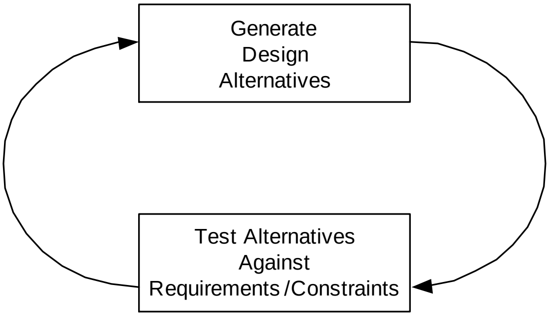


Figure 1. The generate / test cycle (Simon, 1996).

Consistent with the design cycle described by Simon, this work also consists of different phases refining the design process. It starts by using the information gained from the earlier Java Card 3 prototype, which gives the basic viewpoint for this research. This information provides the starting point for the conceptual framework described by Nunamaker et al. (1990). Next the requirements are gathered and the architecture for a credential model is formed using a literature review. With that information the Java Card 2 prototype is designed and built to test the credential model with a real hardware platform. Finally, the constructed prototype is tested further by implementing an additional ticketing service on it; although this service does not use the same credential protocol, it does use the same credential storage method without modifications to the prototype applet.

3. Background

This chapter describes the basic background components behind this study. They are identified based on the previous literature and from the earlier prototype. The viewpoint is on usable security, and thus the theoretical concepts behind it are defined first. The use case involves web technologies, and the associated security-related threats are described. These include browser-based threats such as password phishing and eavesdropping attacks. Although not strictly in the scope of this work, they are central concepts that need to be defined when discussing password management strategies. Lastly, concepts of identity, privacy and service management are described. They are key components that define how the prototype architecture described later in this work is built. Also, basic concepts relating to the hardware platform – like secure elements, servlet environment and web technologies – are defined.

3.1 Usable security

The chain of security is only as strong as its weakest link, and the weakest link in this case is often said to be people. Security is often understood only from the technical perspective and its human factors are neglected. These considerations can also be seen to be mutually contradictory, meaning that security is designed at the expense of usability and that this cost is accepted. However, this kind of mental model of security creates unrealistic expectations for the user (Cranor & Garfinkel, 2005). Users see security measures as an additional burden and rarely have the motivation to comply with them. People tend to ignore and circumvent security measures where possible (Weirich & Sasse, 2001). Also, due to the great usability issues in existing software, users avoid adopting software such as cryptography tools (Whitten & Tygar, 1999). Chiasson et al. (2006) researched the usability of two password managers and found out that they generally are not a good fit with the users' mental model. This causes severe security problems, as users do not necessarily know when a password is protected and when it is not. It creates a false sense of security for users, which might compromise their security. The same kinds of security issues can be found in security toolbars. Wu et al. (2006) conducted two user studies of three security toolbars and found out that many users do not understand the nature of phishing attacks and that these toolbars have many security issues. Alarmingly, users sometimes ignore alerts about a phishing site because the site looks so authentic and real.

Usable security approaches this problem from the user's perspective. Security needs to be constructed so that it is simple enough for the user and based on the correct mental model of the user. One strategy for this is to employ persuasive technology (Fogg, 2002). Persuasive password technologies can persuade the user to use more secure passwords without sacrificing usability (Weirich & Sasse, 2001). Chiasson et al. (2008) have researched this in the field of graphical passwords and found that persuasive password methods create more random passwords that are still usable.

3.2 Browser-based threats

With traditional smart card technologies, the chain of security is usually well known. Trusted smart card devices are used with proprietary trusted reader devices. With web-enabled smart card technologies, such as Java Card 3, the situation changes. The client logic is moved from a proprietary application to a more open environment, possibly to a standard web browser or at least to a client with a significant amount of web connectivity. This leaves room for unknown elements, as the application provider does not control all of the software components. There are two common attacks that can affect a smart card application running in a web-enabled environment: man-in-the-middle-attacks and man-in-the-browser-attacks.

A man-in-the-middle-attack means a situation where the attacker is able to intercept the communication and eavesdrop on it as shown in Figure 2. The attack starts when the attacker intercepts a message to the service originating from the user. The attacker then makes its own connection to the service and proxies messages between the user and the service, listening to and altering them. When communicating with a web-enabled smart card that is attached physically to a computer, a man-in-the-middle-attack could mean that malicious software on the user's computer is intercepting communication between the browser and the card. A man-in-the-middle-attack can be detected and prevented with different techniques such as using HTTPS with keys verified by a common certificate authority.

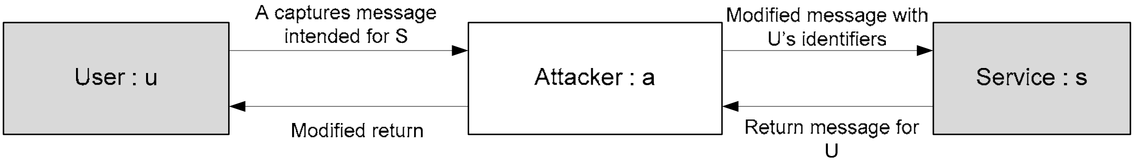

A captures message intended for S
Modified message with U's identifiers
User : u
Attacker : a
Service : s
Modified return
Return message for U

Figure 2. MitM attack.

A man-in-the-browser-attack is possible when the attacker is able to execute malicious code inside the user's browser with browser-helper-objects, browser extensions or direct browser manipulation. Figure 3 shows the basic principle of a man-in-the-browser-attack. This enables the attacker to listen in and modify web pages used in the browser. The attack begins when the attacker infiltrates the browser with some method such as a trojan. The attacker then loads the browser extension that handles every page-load on the browser. When the user enters a specific page, for instance an online banking page, the attacker detects this and modifies the page or communication without the user's knowledge. The attacker then sends the modified details to the web server and, when receiving a response, modifies it with the original values.

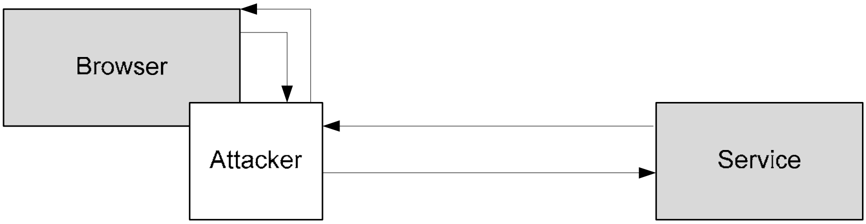


Figure 3. MitB attack.

Man-in-the-browser-attacks are harder to prevent than man-in-the-middle-attacks since all single-channel authentications are circumvented in the browser. Gühring (2007) presents possible solutions for man-in-the-browser-attacks, which include trusted computing and a second secure channel of authorization or external authorization device.

3.3 Password phishing

Phishing is identity theft where the user is tricked into giving out confidential credential information to the attacker. This information can be, for example, passwords or credit card numbers. Phishing attacks can be executed using the classic technique of fraudulent e-mails, but increasingly also with malware and DNS-based attacks. The losses related directly to phishing were estimated to be $1.2 billion in 2003 and indirect losses even greater. (Emigh, 2005).

Numerous experiments have been carried out with different kinds of security toolbars as an answer to the phishing problem (Chou et al., 2004; Gajek et al., 2007; Herzberg & Gbara, 2004). The security toolbar tries to identify whether the user is on a legitimate web site by analyzing server certificates and the domain name. However this is not always without problems, because not all sites use verified SSL certificates (Wu et al., 2006).

3.4 Identity and privacy management

Nowadays users need more and more passwords for different services. Gaw & Felten (2006) documented in their research that a majority of users have only a few passwords and tend to reuse them in different services. Furthermore, reuse rates increase over time, because users accumulate new accounts. (Gaw & Felten, 2006). This creates a so-called “domino effect” where an attacker can gain access to multiple places with one exposed password (Ives et al., 2004). Different kinds of password managers and security toolbars are used as a solution to this problem (Chiasson et al., 2006; Chou et al., 2004; Wu et al., 2006). However, no general solution for the problem exists yet and the current solutions are usually platform- or software-dependent. Some have counterparts that work in the web, but generally they are not usable enough (Chiasson et al., 2006).

Users network more nowadays than ever before and even share personal information more widely. It is not always clear how and for whom the personal information is available. Gross et al. (2005) found out that people share information about themselves willingly even though only few use available privacy control mechanisms in order to limit who can view this information. Researchers were also able to link profiles between different services using the pictures listed in the profiles. Nissenbaum argues that the line where people feel their privacy is violated is defined by contextual and temporal integrity in a given situation. What people think is appropriate in one relationship might be inappropriate in another. (Nissenbaum, 1998). On the other hand, privacy violations are not always intentional (Adams & Sasse, 2001). In an organizational context, it has been proposed that an organization should have an integrated solution to protect the privacy of data from not only outsiders but also people inside the organization. According to Brodie et al. (2005), privacy functionality must also be separated from application code for reasons of cost, consistency and flexibility and it must also support an appropriate level of granularity. For most of the services on the Internet, there is no central way to control privacy levels, and privacy controls are usually vague and inconsistent.

3.5 Secure element and trusted computing

A secure element is a computer platform that has its own processor and memory and is both physically and programmatically secured against tampering. A secure element can work alone with the help of a reader device, like in credit cards, or can be embedded in another device, like in computers and some mobile phones. The most widespread secure element hardware is Java Card, with more than three billion smart cards deployed worldwide. It enables smart card chips to run special Java applets designed for the Java Card platform. The platform includes a private partition for each applet on the card. Communication between applets is shielded with a firewall controlling shared data. Object ownership change or a special Shared Interface Object (SIO) can be dedicated for inter-applet communication in a secure manner. With SIO, the implementing applet can define a normal Java interface that can be used by other applets to provide access to protected information. Applets run on top of a virtual machine, separating them from the hardware and card operating system. Applets are also signed with a cryptographic signature and can therefore be securely distributed to third-party developers. The recently released Java Card 3 platform also includes a small web server that can communicate with standard Internet protocols, such as HTTP and HTTPS. Web server capabilities enable implementation of communication protocols with Representational State Transfer (REST) interfaces similar to many currently existing web services. In the Java Card 3 platform, Java applications run as web-enabled servlets and can also act as web clients through Java's Generic Connection Framework (GCF). An overview of the Java Card 3 platform architecture is given in Figure 4. (Sun Microsystems, 2008).

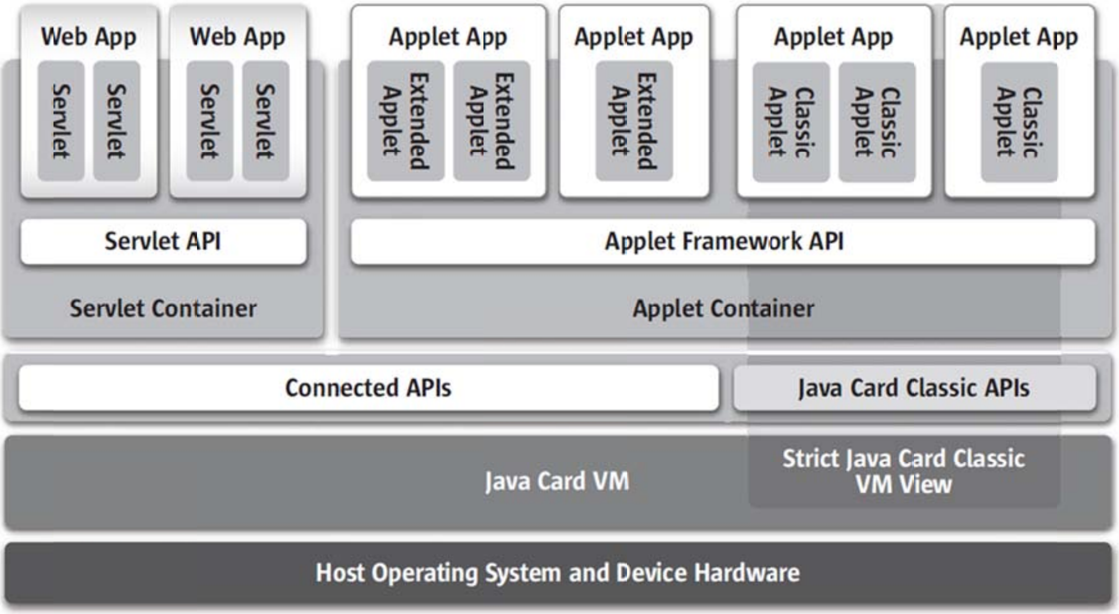


Figure 4. Java Card 3 architecture (Sun Microsystems, 2008).

Another widely deployed platform for secure computing is the Trusted Platform Module (TPM) developed by Trusted Computing Group, a not-for-profit industry-standards organization aiming to enhance the security of the computing environment. TPM is developed to provide signing and cryptographic services as well as an environment for trusted execution of applications. The term "Trusted Computing" is taken from the field of trusted systems. Key concepts in the TPM specification include the endorsement key, a security feature that is initialized inside the chip only once and provides proof of origin for the chip user, and the attestation of messages, which means verification of message origin with cryptographic keys. Capabilities of TPM chips also include memory curtaining and protected execution inside the chip. Memory and processes are shielded from both outside processes as well as other processes inside the chip. TPM can be found for instance in many modern laptops as well as in some desktop computers and tablets. (Trusted Computing Group, 2003).

ARM TrustZone is a technology used in modern ARM processors to offer a secure execution environment. TrustZone is based on a system-wide security environment where all applications can be run in either normal or secure mode. In the processor architecture, two virtual cores are used, one for non-secure execution and the other for secure execution. System resources are limited so that the non-secure core can only see non-secure resources but the secure core can see all resources. Execution between virtual cores is done in time-sliced fashion through a core mode called the monitor mode. System-wide security also extends to other parts of the system such as MMUs that also have virtual parts for each virtual processor securing memory from non-secured code. An overview of ARM TrustZone architecture is shown in Figure 5. (ARM, 2009).

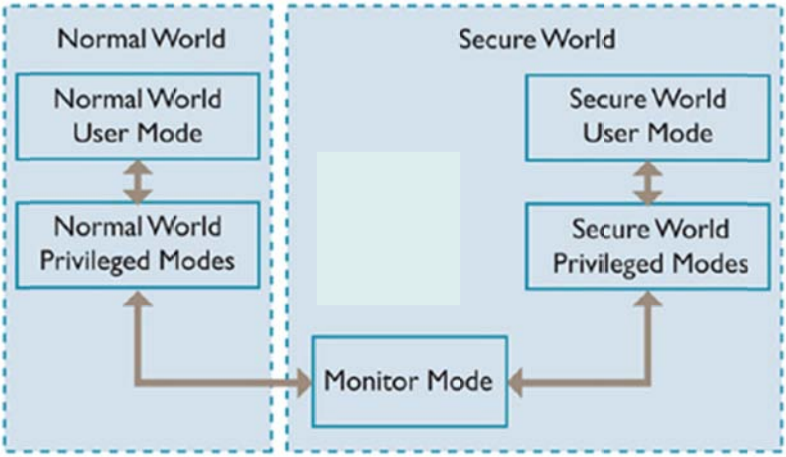


Figure 5. ARM TrustZone architecture (ARM, 2009).

3.6 Smart card hardware

Smart card research dates back to 1968 when the idea of plastic cards with microchips was first patented. However, the cost of chip manufacture declined to reasonable levels only after 1976 and commercial applications started to spread in the 1980s (Husemann, 1999).

Smart card applications are now widespread and we use different smart card applications everywhere. The most common are different kinds of credit cards, SIM cards and both access and payment cards. Several different specifications standardize card details, but many cards nevertheless differ from each other. The most manufacture-specific part in current cards is the card operating system, which is usually proprietary even if the card itself is equipped with a Java Card virtual machine. The basic concepts of smart card hardware architecture and the specifications controlling it are described in this chapter.

3.6.1 Architecture

There are basically two types of smart cards: Memory cards and microprocessor cards. The simplest type of smart card is a plain memory card without a processor, which is used in services such as public transport ticketing. Memory card-based chips usually have only a simple circuit with a few pre-programmed instructions and no programming capabilities. They are also more vulnerable to counterfeiting and tampering than smart cards with a processor. Microprocessor-based cards are far more flexible and secure, because they can be reprogrammed or updated. Almost all of the currently available cards also use certificates to protect applications and prevent installation of unauthorized applications. A smart card processor usually operates at a speed of only a few megahertz and has only tens or hundreds of kilobytes of writable memory. (Chen, 2000).

3.6.2 Specifications

The implementation of smart cards, readers and applications is controlled and defined by several different standards that specify physical and application-level characteristics. The most important and widely used standard is ISO 7816 “Identification cards – Integrated circuit cards with contacts”, which defines the physical characteristics as well as application identifier and the interface between the card and the reader. The ISO 7816 standard applies to all contact chip cards. Use of chip cards in mobile phones is standardized by the Global System for Mobile Communications (GSM) standards defined by the European Telecommunications Standards Institute (ETSI). These standards define the characteristics of SIM cards and their application toolkit properties. (Chen, 2000).

Another standard covering the entire smart card infrastructure is GlobalPlatform, which is implemented in many of the currently available smart card-related devices. It defines specifications for cards, readers and systems in an industry-neutral way. Most of the specifications are freely available and go through a three-staged review process, during which the documents are initially reviewed by members of the GlobalPlatform organization committee. After committee review, each document is reviewed by GlobalPlatform members and then undergoes a public review. (GlobalPlatform, 2010).

For reader interoperability, the PC/SC Workgroup has set PC/SC specifications that define the architecture for smart card usage in personal computer systems. In PC/SC architecture, host-side smart card applications are built on top of one or more service providers and a resource manager. These expose smart card-relevant programming interfaces and resources within the system. PC/SC shares many concepts with another framework middleware: the OpenCard Framework (OCF), which was owned and developed by the OpenCard consortium, but has since then split up. (Chen, 2000).

The EMV standard is a specification intended for the needs of the financial industry. It is defined by Europay, MasterCard and Visa and it extends the ISO 7816 standard with proprietary features intended for payment applications on smart card and payment terminals such as point-of-sale terminals and ATMs. (Chen, 2000).

3.7 Servlet environment

A servlet is defined as a web component based on Java technology. It is managed by a container or the servlet engine that generates dynamic content. Servlets are platform independent and executed by a Java-enabled web server. A servlet is implemented with a request/response model, where each query is modelled as a request and dynamic content is served via a special response object. Servlet requests are mapped through the application name defined in the servlet manifest. (Sun Microsystems, 2009).

In the Java Card platform, authentication to servlet is done with HTTP/1.0-compliant HTTP Basic or HTTP Digest authentication. The third possibility is form-based authentication, which provides the look and feel of a login screen. However, web-based login suffers from some weaknesses compared with a traditional Java Card reader and PIN. With HTTP Basic and form-based login, the password is transmitted as plain text; without additional security measures like HTTPS or VPN, the connection can be vulnerable to man-in-the-middle-attacks. (Sun Microsystems, 2009). Additionally all authentication methods are vulnerable to man-in-the-browser-attacks if the browser is not secured with security measures.

3.8 Open credential protocols

OAuth is a credential protocol for web services that provides a way for a third party to access server resources without revealing the actual credentials such as the username and password for the service. The credential protocol works with user-agent redirections by redirecting the user to a service where user authentication is used to generate an authentication token specific to a certain third-party service. After generation, the token can be used to sign all requests for the server resources until it expires or is revoked. (Hammer-Lahav, 2010).

OpenID is intended to provide the end user with control over the authenticator credentials. They are provided by a nominated third party and supported by the web service. OpenID implements many advanced features such as identity service discovery (Recordon & Reed, 2006). OpenID and OAuth can be combined to provide a combined shared identity to grant third-party access (OpenID Foundation, 2010).

4. Credential platform requirements

This chapter identifies and defines the requirements for the credential platform based on the literature about previous research. This is done in three parts: Password management theories, existing credential platforms and interoperability requirements for the implementation environment. Password management theories are selected for investigation in one part of this review in order to effectively identify the user actions in the authentication process from the usable security viewpoint. These theories concern privacy and trust management, but the focus of this work is only on password management. Credential platforms are researched to identify existing solutions to credential management in a trusted platform. Lastly, interoperability requirements are identified to create a platform-independent system.

The first problem area is managing a user's credentials, such as username, passwords or credit card information. Password management strategies are a well-researched field and studies show that users tend to use short and simple passwords; if they are forced to use strong ones, they usually reuse passwords in different services. Attackers use this knowledge by attacking services that are under weaker protection and try to use the cracked passwords to access stronger services. (Gaw & Felten, 2006; Keith et al., 2007; Yan, 2004).

Credential management can be understood to involve both managing a user's credentials for a service provider as well as managing service provider identities like SSL certificates for the user. This work uses a credential store as a solution for virtual single sign-on implementation. This is similar to the user-centric identity management model presented by Jøsang and Pope (2005).

A formal literature review is conducted to research password management theories and existing credential platforms, whereas interoperability is reviewed more informally to keep the scope and size of the literature review coherent.

4.1 Literature review

The target of the literature review is to answer the question: How are existing credential management systems implemented and what needs to be considered based on background theories related to credential and password management? The review is conducted by deriving categories and keywords that are specific enough to yield information about background and implementation. This is done by analyzing the found literature before proceeding to the actual literature review.

In the first phase of the review, an overview of the study field was acquired by means of a traditional literature search. The Google Scholar search engine was used to obtain as many sources for data as possible. Promising sources were also analyzed for additional bibliography. Based on the context of sources, the following main categories were identified: Password strategies, privacy and trust management and credential design. Credential management is closely related to the design process where core design choices emerge from the identity and credential models, and they are thus central categories when creating the design. Also, password strategies and privacy and trust management models are central concepts when designing the identity and credential models, because they explain user behaviour related to password usage. Privacy and trust management was combined with password management strategies later on in this review and not examined separately to keep the length of the review manageable. However, this field was not completely disregarded since it also contains many password management-related articles.

After categorizing sources, keywords were identified. These keywords are listed in Table 1. Some articles already included defined keywords, but most did not. For those articles that did not have keywords, central words were picked from the abstracts. The collection of keywords was fine-tuned further based on a pilot search through databases. The table below presents common keywords and databases based on the initial search. All keywords are expected in articles except “user behaviour”, “user-centred” and “mental models” in password strategies, which were used analogously to each other in the initial search. Two of the most popular databases relevant to the literature review are ACM and IEEE. ACM is especially relevant for password- and privacy-related research, since it contains a wide range of usability and human-centric research. IEEE is relevant for credential design, since it has a wide range of technical research available. The initial search supports the selection of both databases. In addition, SpringerLink was selected as a source, as it is one of the central databases in this research area. Only these three key databases were selected to keep the amount of results manageable, as the literature review is only part of this work.

Table 1. Discovered keywords in initial search.

Category	Keywords	Databases
Password strategies	Security AND password AND usability AND (user behaviour OR user-centred OR mental models)	ACM, IEEE, ScienceDirect, Springer-Link
Privacy and trust management	(privacy AND usable security AND trust)	ACM
Credential design	(authentication AND trusted computing AND smart card AND design AND credential)	Springer-Link

The purpose of the review is to find implementations of credential systems for trusted computing platforms. Therefore the criteria for selecting the papers for review are: the existence of a trusted platform, implementation of a credential system or credential scheme based on it, and usage of the system in some type of authentication. The use of these criteria in the literature search uncovered research on a wide range of applications, but the topic that all the articles had in common was research on credential management schemes, which is needed to build a secure and functional system in this work. The literature search reveals in total 633 articles, 104 of them focusing on credential systems. Articles were first selected based on the title and abstracts. The introduction and conclusion of the remaining articles were further analyzed in order to select the final set of articles for closer review.

Table 2. Number of articles discovered.

	ACM	IEEE	Springer	Total
Password strategies	37	157	97	291
Privacy and trust management	116	91	31	238
Credential design	17	56	31	104
				633

15 of the 104 articles focusing on credential systems were selected for closer review. The criteria for selection were the construction of the system and the use of a trusted platform. The disregarded articles concerned trusted computing or credential platforms, but were either not constructive by nature or the construction implemented was not within the trusted platform, but rather more in the external infrastructure. The 15 selected articles were classified and analyzed based on their credential design models. The articles were classified based on whether the credential system is intended for multi-application or multi-purpose credentials and whether the application is user or issuer manageable. User manageable systems also include those that are manageable by the users themselves through a third party or the issuer. The problem field and proposed solution were also identified and this information is used as the basis for credential design later on in this work. A detailed description of the credential system-related articles is presented in the following table.

Table 3. Papers describing adaptable credentials.

	Single application	Multi-application	User-centric (open)	Issuer-centric (closed)		
Paper	Credential model		Ownership model		Problem field	Solutions proposed
Akram, Markantonakis & Mayes, 2009		X	X		Open multi-application card	Open card based on TSM managing card applications.
Alzomai & Jøsang, 2010		X	X		OTP-based access to multiple services	TPM-based OTP generation scheme in mobile phone.
Bichsel, Camenisch, Groß & Shoup, 2009		X	X		Anonymous identities	Efficient anonymous attestation model.
Brickell & Li, 2007		X			Anonymous identity	Anonymous attestation model with TPM
Gardner, Garera, Pagano, Green & Rubin, 2009	X			X	Medical record secure access control	Shared secret model allowing different access methods
Hyppönen, 2008		X	X		Open electronic identity	Identity proxy based on secure element in SIM. Key pair architecture.

Kalofonos & Shakhshir, 2007	X		X		Smart home access control and delegation	Mobile phone-controlled smart home credential system including access delegation
Klenk, Kinkelin, Eunicke & Carle, 2009		X	X		Identity theft	Trust establishment scheme with smart card, TPM and OpenID.
Khan & Hayat, 2009	X			X	Governmental identity management	Identity management and attestation framework for governmental services
Li, Zhang, Seifert & Zhong, 2008	X			X	Mobile payment	Flexible and efficient mobile payment architecture using TPM
Maher, 1998		X	X		Electronic currencies	Support for dynamic currency definition with financial cryptography protocols
Sandhu & Zhang, 2005		X			Trusted peer-to-peer architecture	Security policy protocol for trusted platform
Toegl & Hutter, 2010	X			X	Integrity attestation	NFC-TPM-based attestation protocol
Bauer, Cranor, Reiter & Vaniea, 2007	X		X		Physical access control	Rights to delegate access tokens forward to other users and groups
Ekberg, Asokan, Kostiainen & Rantala, 2008		X	X		Multipurpose open credential platform	Attestation platform for trusted computing allowing multipurpose credentials with different access groups

The reviewed articles use different credential models, but there are four common categories: Attestation, access delegation, multiple credential models and anonymous credentials. Since the scope of this work is services where the identity of the user is known and bound to the service itself, anonymous credentials are not as relevant as other types. Also, attestation is included in other models and therefore the delegation of access and management of multiple credentials are the most relevant issues for the scope of this work. A common trend in the reviewed articles focusing on shared credentials is to use a user-centric credential model. This is usually implemented by allowing the user control over what services to include on the card or how to delegate access to different users. This makes sense in the context of usable security, since there are usually a great number of services and their configuration varies from user to user. With the single application model, the credential model is usually issuer-centric; the reason for this is that applications require control over credentials. Applications include payment-related applications (Li et al., 2008), governmental identities (Khan & Hayat, 2009) and medical record storage (Gardner et al., 2009). It is also noteworthy that the open user-centric model used in the reviewed articles differs from the current model used in many types of web service authentication where individual credentials are strictly controlled by individual services and allow very little flexibility with regard to different management models such as persuasive strategies or single sign-on.

The second part of the literature review comprised the collection of information about password management theory. The literature search initially covered theory concerning password management and trust and privacy management. Of the 633 articles revealed in the initial literature search, 529 were on the topic of password, privacy or trust management. Due to the scope of this work, only articles relating directly to password management were analyzed more deeply. The articles were also required to evaluate password management strategies from the user's perspective or mental models, which excluded the majority of articles focusing on the technical and implementation level of password management policies. A total of 11 articles were selected for deeper review from the password management articles based on the article abstracts. The low number of selected articles is explained by the fact that the search criteria also covered privacy- and trust-related articles, which brought up a great number of articles that did not concern password management strategies. However, their inclusion was useful, since half of the included articles are from that category. An article analysis is presented in the following table.

Table 4. Literature review on password strategies.

Paper	Research question	Password management strategies
Adams & Sasse, 1999	Study of user mental models	Enough training must be provided for users, along with additional feedback during the password construction process in order to make users construct passwords that do not circumvent security mechanisms. Users must also be made aware of different security aspects and their importance in the organization and possible threats against them. Also, the number of passwords must be minimized to four or five unrelated passwords.

Paper	Research question	Password management strategies
Forget, Chiasson, van Oorschot & Biddle, 2008	Persuasive strategies to influence user password management models	Users are not willing to comply with or spend time on security measures if it takes away time from their primary task. Persuasive authentication attempts to solve this by personalizing the authentication procedure, making it as simple as possible and subjecting the user to monitoring and conditioning to encourage correct behaviour.
J. B. Gross & Rosson, 2007	Qualitative interview study to find out user perceptions about security concerns and the party responsible for security management	Users are somewhat aware of existing problems, but their information is outdated and misinformed. Users might also have good habits in physically protecting the computer by locking it and keeping passwords private, but might nevertheless delegate more responsibility for security considerations to IT staff in organizations and have trust in them and their colleagues. One reason for this is that users do not feel comfortable with their own skills.
Herley, 2009	Motivation of users' security measures through the lens of cost	Herley proposes that users are not simply unmotivated to improve their security behaviours, but rather that the cost in working time is too high for users when compared to monetary losses. Even more so, because monetary losses are partially or fully covered by the service provider or other parties. This time loss is a more direct and instant effect compared to the cost of security attacks and makes users unwilling to follow security policies.
Horcher & Tejay, 2009	Role of cognitive load and password training in the password selection process	Horcher and Tejay propose that password selection training given in the form of simple achievable rules reduces the cognitive load of the user and helps the user to create more secure passwords. A user experiment comprising initial password testing, training in form of a quiz and analysis of the created quiz is conducted.
Inglesant & Sasse, 2010	Password usage study on password policies and the user	Users feel that organizational password policies are a burden and tend to circumvent them. Policy circumvention creates security risks when users write passwords down. It can also create additional workload when users start to forget their passwords often and need to reset them.
Mannan & van Oorschot, 2008	Phishing and password study on online banking and security	Users and banks have different expectations concerning security requirements and most users seem to fail the security expectations set by banks. Users also have a vague mental model about the required security measures or concepts like URL checking for phishing sites.

Paper	Research question	Password management strategies
Singh, Cabraal, Demosthenous, Astbrink & Furlong, 2007	Social and cultural dimension in security design for online banking	According to Singh et al. (2007), the social and cultural dimension of online banking creates a gap between users and banks. In Asia and developing countries it is a common practice to use public computers or share mobile phones. It is also a common practice to share passwords between family members.
Singh, Cabraal & Hermansson, 2006	Social dimension in security design for online banking	Singh et al. (2006) propose that the social dimension must be taken into account in security design in order to design systems whose users can and will comply with security requirements. In online banking, users tend to share passwords between family members, which creates a legal gap between the banks' security behaviour and the users' social practices.
West, 2008	User security-related actions based on user motivation and risk	According to West, users are not motivated to perform necessary security tasks. This is because the perceived gains are lower and not as immediate as when the user is taking a known risk that yields immediate gains. Users also perceive that they are less at risk than other people.

The viewpoint in this literature review is on usable security. This issue is clearly investigated in the analyzed articles. They offer a multitude of solutions for how to get users to create stronger passwords and to follow security guidelines. All of these solutions are related to the users' mental model and how to achieve a better fit with the way in which users normally use passwords. Strategy must be more persuasive than restrictive and fit into the users' social practice.

The perspectives taken in the analyzed articles are different – and sometimes even partly opposing – but all the articles have the same message. The reason why users do not follow security requirements is not that they are lazy or because they do not want to. Instead security requires the user to expend too many resources in terms of time, money, concentration or anything else the user needs to focus on in his or her main task. Password management also creates a cognitive load for users that must be reduced in order to allow them to create more secure passwords. (Horcher & Tejay, 2009). Security is perceived as a secondary objective that does not produce immediate gains. Instead, it has immediate drawbacks for the primary objective. In the literature review, it is also clear that users will not follow guidelines that are vague, too strict or deviate from their social practice. Instead they go around the restrictions by reusing passwords, using weak passwords or by sharing passwords. By neglecting security, users put themselves and systems at risk. However, users perceive the risk to be smaller than it actually is and themselves as being less vulnerable than others. Therefore, the constructed application must produce additional value by simplifying the authentication procedure and by fitting into users' existing mental model and work practice. Simply forcing users to follow rules and guidelines is not enough, as users tend to bend them. (Forget et al., 2008; Herley, 2009; Inglesant & Sasse, 2010; West, 2008).

The articles propose a number of solutions that are suitable for the smart card environment constructed in this work: minimizing the number of passwords, as suggested by Adams and Sasse (1999), ensuring the transparency of the authentication procedure in order to lighten the users' workloads and give them time to focus on their primary objective, as proposed by Herley (2009) and West (2008), and making a tool that provides a better platform for implementing password creation with persuasive strategies, as discussed in the article by Forget et al. (2008).

On the other hand, too much simplification leads to the risk of compromising security. If all credentials – from frequently used social media applications to credit cards, where high security is essential – are shielded with one password, the attacker only needs to obtain that single password to gain access to every piece of information. This creates the need to achieve a careful balance between security using a persuasive usable approach and an unusable amount of different credentials. One possible solution for this kind of problem is a password manager or security toolbar. The problem with these software implementations is that they usually involve usability problems relating to the user interface or the user's mental model for the service. The user does not necessarily understand when a password or page is protected and when it is not. Other problems relate to password storage and platform interoperability. The password manager is usually bound to the user's browser and is not easily accessible from different computers. This creates problems if the service passwords are changed and the user does not know or remember them. (Madlmayr, 2008; Wu et al., 2006).

Virtual single sign-on, as proposed in earlier studies, allows the user to access different services with one or several passwords (Klenk et al., 2009). This type of identity management can also be implemented using an open model with a secure element. This allows extendable services that the user can configure as required. (Akram et al., 2009; Alzomai & Jøsang, 2010).

4.2 Interoperability requirements

Interoperability between secure element platforms has usually been poor. Different platform vendors use different trusted platforms that are not interoperable. Even between Java Cards implementations have varied so much that mobile operators have had to virtually redevelop software from the ground up when changing over from one card manufacturer to another. If the user experience with the application stays the same between different platforms, the application has a higher probability of being more usable. Therefore a platform that is more interoperable with current devices is sought and it is tested whether this would yield higher usability. Nowadays interoperability has improved greatly, but developers still need to write special client applications in order to communicate with the cards. It has been proposed that there should be a standardized protocol supporting networked communications. (Vandewalle, 2005). Java Card 3 offers the standard HTTP protocol as a solution for this problem. Solutions have also been launched for the Java Card 2.2 platform. For instance, Smart Card Web Server offers a web server platform for Java Card 2.2. (Open Mobile Alliance, 2008; Sun Microsystems, 2008).

It is not a straightforward process to create a credential management platform that is usable and available in any situation, even when the used hardware platform changes, and which also remains secure enough to gain the user's trust. The IETF Securely Available Credentials (SACRED) working group has discussed this kind of implementation. The working group describes two possible solutions: credentials distributed directly from device to device and a credential server (Farrel, 2004). Both solutions have their problems. In a server environment, users need a trusted service manager that handles credential storage. Data must be always available and stored securely so that not even the service provider can see the user's credentials. Trust is usually linked to the uncertainty and risk related to the service (Kelton et al., 2007). Storing vast amounts of highly sensitive information such as credentials for online banking or e-mail accounts increases the risk related to the use of that service. In that kind of environment, users must have very high trust in the service provider. On the other hand, users generally have high trust in device-to-device implementation, like a smart card platform, and in smart cards that enable the users to easily control what information is stored and how it is used. Data availability also differs depending on the implementation. On the other hand, a credential server can be used everywhere over an internet connection. However, it is vulnerable to network outages whereas device-to-device implementations are available whenever a device is available and connected. Also, if a credential manager is placed on a mobile device, it should provide proper backup functionality and preferably recovery features in case the password is forgotten, and do so without compromising security.

The system design and hardware platform should give the user the choice of using the system wherever it is needed, giving the user a similar user experience with all devices. In the best case, this could mean standardized communication protocols, such as HTTP, with minimal driver support needed. With current hardware limitations and availability in mind, interoperability cannot be formally evaluated from all viewpoints, but this work creates an overall view of the issues that need to be considered when creating a smart card application for managing user credentials.

4.3 Summary of the requirements

The results of the literature review are summarized in this chapter. The following list summarizes the results in four central categories that represent the analyzed material from the viewpoint of this study. The central categories are: multiple credentials, secure passwords, ease of use and interoperability.

Multiple credentials

- The credential system must be open and user controllable.
- The user must for instance be able to select what services to use or how to delegate access.

Secure passwords

- In order to make the user follow security guidelines and create better passwords, the cognitive load of the user must be minimized by reducing the amount of needed passwords. Some of the reviewed credential models use virtual single sign-on to achieve this.
- The password model must fit into the mental model of users to help them to use passwords securely and correctly.

Ease of use

- Credential usage must be transparent, thus reducing users’ cognitive load.
- Users must be able to tell when a password is protected.

Interoperability

- The user experience must stay the same with different platforms.
- The secure element must support a wide variety of platforms with minimal driver requirements.
- The application must be able to support credential backups and synchronization in a secure manner.

These results serve as requirements for the design in the next chapter. They are the major issues that must be kept in mind during system construction and serve as a basis for the evaluation criteria later on.

5. Prototype design

This chapter presents the design choices for the prototype constructed in this work. The design choices are based on the requirements derived from the literature review and existing credential designs. A generic single sign-on protocol is selected as the main design choice, as this protocol has been used in earlier literature and it fulfils the requirements for the multiservice application on a smart card. It is also a feasible choice for a smart card as it creates a lightweight generic framework that requires little support from the smart card applet to support new services. The downside is that a smart card can only support services compatible with the credential protocol, but this is still adequate for this work, enabling the testing of the suitability of a web-connected multiservice model for a smart card.

5.1 Prototype definition

After the definition of the design model, prototype implementations are described. Prototype implementation is done in three parts. Prior to this work, a Java Card 3 emulator implementation of the prototype was used to test the basic concept and provide the viewpoint for this work. The prototype implemented in this work is a simplified version of the earlier prototype implemented for a microSD-embedded Java Card 2 smart card on a mobile phone. The earlier prototype is used to create the viewpoint for this study and to test the functionality of the concept. The purpose of the prototype in this work is to validate the concept in currently existing hardware against the requirements defined in the literature review. In addition, a small ticketing service is implemented on top of the prototype. The ticketing service is not actually fully in the scope of this work, but provides a viewpoint on how the existing credential storage can be used to implement a service that does not follow the original authentication protocol, but can still share the same credential storage without modifications to the credential applet. A detailed description of the additional service is provided as an appendix to this work.

5.2 Prototype platform

The hardware platform used with the earlier Java Card 3 prototype was a high-end desktop computer with Java Card 3 SDK. The prototype was implemented on an emulator, which does not require a considerable amount of computing power. The emulated card was a very high-end card with 4MB of RAM and 8MB of writable EEPROM, although the emulator numbers are not comparable to actual cards.

The Java Card 2 prototype in this work is implemented for the generally available Java Card 2.2.1 platform. Along with the Java Card version, the most critical difference between the prototypes is that the Java Card 2 prototype is implemented on a mobile phone with a Java Card embedded in a microSD memory card. The microSD used in the prototype is manufactured by GO-Trust and has an embedded 2.2.1 Java Card chip with support for the GlobalPlatform standard along with 1GB of SD memory; bigger SD memory is also supported. The mobile phone used for the demonstration is HTC Desire with Android 2.2. Since its drivers also support J2ME and WinCE implementations, other implementation platforms besides Android phones would be possible.

For interoperability, the prototype applet was also tested with Mobile Security Card, manufactured by Giesecke&Devrient. Version 1.0 of Mobile Security Card contains Java Card 2 with GlobalPlatform support and 2GB of SD memory. Drivers support Android and currently require a custom compile of Android to operate.

5.3 Java Card 3 prototype

The first prototype was the predecessor to this work. It was done in a project at VTT and its purpose was to find out whether a single sign-on application with user controllable services can be implemented with Java Card 3. The role of the first prototype is to provide perspective for this work and the second prototype. Because Java Card 3 generation cards are not yet available on the market, the prototype was implemented on an emulator and card contact with the reader was also emulated with a notifying application. This enabled the application to act as if a real card were used with a reader.

The prototype was targeted at desktop use, but mobile use was also kept in mind. An NFC phone was used to mimic a card along with contactless smart cards. As Java Card 3 works as an HTTP server, it enables versatile communication between the applet and client. All communication between the client and applet works with HTTPS protocol, making the applet very interoperable and platform-independent.

5.3.1 Supported services

The implemented application supported social media and e-mail services. In addition, some mock-up applications were used to demonstrate a multi-application card with services such as access cards and credit cards, which would also benefit from a user manageable credential model as Bauer et al. (2007) demonstrated with a user manageable access-control system.

All the Internet services implemented on the card used OAuth or similar to authenticate and provide information through different REST interfaces. Internet services included Google Gmail, Twitter and Facebook feeds. The card’s own services included web-based configuration of different profiles, services and credentials. Information provided from card services was provided in XML and JSON formats in addition to the HTML-formatted interface. Content data from the different services was in a service-defined format without modification.

5.3.2 Credential model

The credential model used in the prototype is an open model, where services can be dynamically added and configured by the user and additional third-party services can be added to the card. The addition of third-party services is implemented with the key management model in the Java Card 3 platform and shareable interface objects (SIO) that act as an inter-application communication method between different applets on the card. Both the key management and the SIO model provide a flexible base for the extension of the applet with third-party services. However, it is noteworthy that in existing applications with smart cards, key management policies and trust relationships between different parties have traditionally been very tight. The device keys are usually very restricted and not shared between different parties, and for this reason service extension is limited to the configuration of the existing services supported by the system, and third-party extensions are beyond the scope of this work. Service configuration was implemented with flexibility in mind. Number of services was not restricted and each service could have multiple credentials acting as different identities. Identities were linked to multiple roles representing different sets of credentials in services, as shown in Figure 6. In a real-life scenario, this means for instance separate home and work e-mail accounts in the system. User authentication to the services is delegated to card credentials that are configurable up to the limitations of the card. For instance, long passphrases, several access levels or biometric authentication could be used. (Sun Microsystems, 2009).

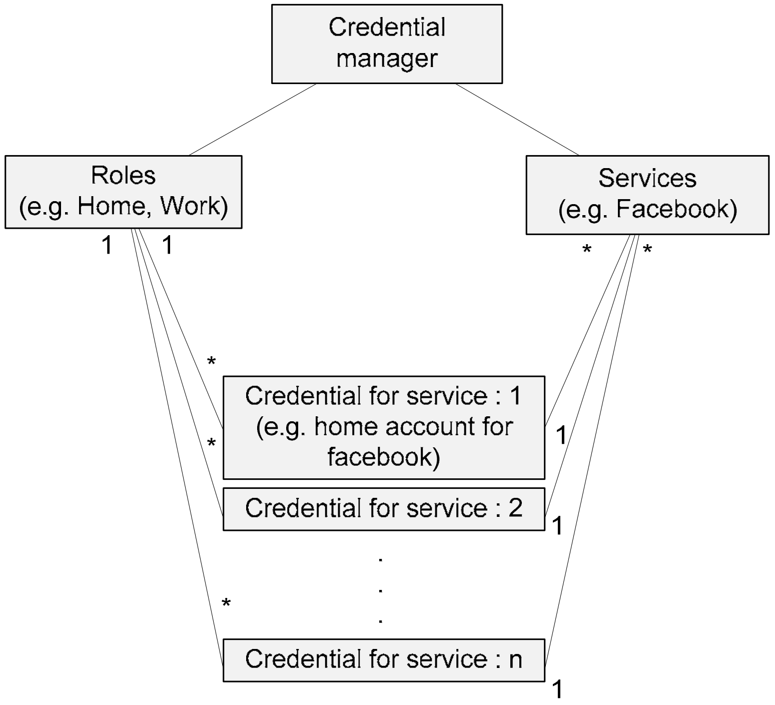

Credential manager
Roles (e.g. Home, Work)
Services (e.g. Facebook)
1
1
*
*
*
*
*
Credential for service : 1 (e.g. home account for facebook)
Credential for service : 2
Credential for service : n
1
1
1
. . .

Figure 6. Credential model for the first prototype.

5.3.3 Architecture

Some services, like Google for instance, use OAuth for authenticating the user. Support for OAuth was implemented as a generic framework on the card, which enables very light implementation of new services on the card. In the best-case scenario, no additional code is required for the new service. In addition, the prototype can be used for proxy API requests from the client to actual services, since all communication is done with HTTP. For the client smart card with API, the proxy is seen as normal API data from services that can be requested from the unified HTTP API on the card. Authentication is done in the smart card, implementing virtual single sign-on. Figure 7 shows the structure of the components and communication interfaces on the card.

Due to the more versatile platform in Java Card 3 and HTTP connectivity, much of the functionality can be implemented on the card side. This enables a simpler and more interoperable client application and enables third-party programmers to easily create new client applications without compromising the security of the applet on the card. On the other hand, it also creates challenges to flexibility, as the applet environment is very strict.

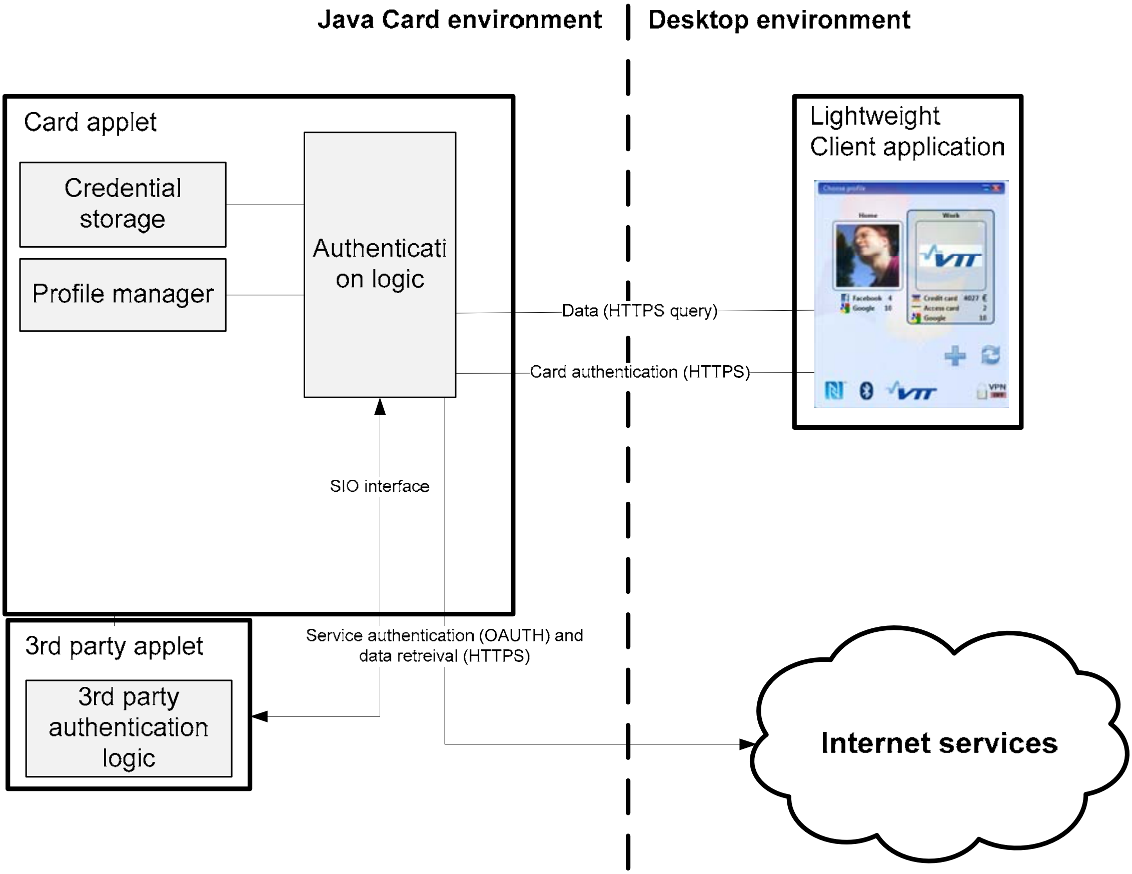
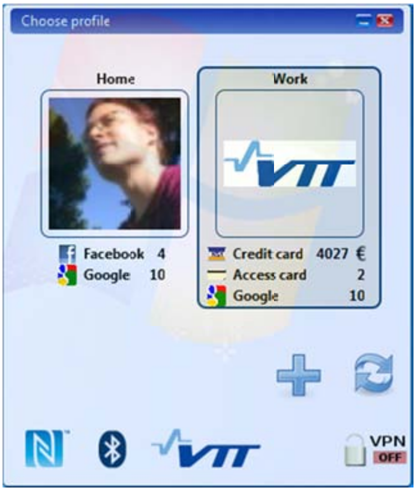


Figure 7. Java Card 3 prototype architecture.

In this architecture, the applet works as an authentication proxy, delivering content data from the web service to the client application by doing OAuth-based authentication transparently on the card. Before using the data, the client does card authentication with the credentials provided for the card. Card authentication is executed only once when starting the client application, providing transparent virtual single sign-on for different services. Service data is provided unaltered for the client application, but the applet has the possibility to modify or reformat it, if needed.

5.3.4 Client application

The client application was implemented with JavaScript and Mozilla XUL. The purpose of the client is to display user interface data and to request content data from the card. The card applet acts as a proxy delivering content data from the service and implementing authentication functions on behalf of the client application. This makes the client application lightweight and easy to replace, as it does not contain any logic relating to authentication. Figure 8 shows the main interface of the client, containing the profile information and individual service accounts linked to it.



Facebook 4
Google 10

Credit card 4027 €
Access card 2
Google 10

Figure 8. Screenshot from prototype client application.

An additional interface for the client is a web interface for managing service connections and profiles. The web interface is generated on the card and displayed through a standard web browser and no additional client application is required. Connection to services could be implemented in the same way.

5.4 Java Card 2 prototype

The purpose of the second prototype is to create a user-centric credential model on a mobile phone based on the requirements ascertained from the literature review. The credential model is refined based on the requirements identified in the literature review and also based on platform limitations. The credential protocol is OAuth, which was also the case with the first prototype, as the literature review also supports the use of an open credential protocol.

5.4.1 Services

OAuth implementation for Gmail. The OAuth protocol implemented on the prototype was used to create a reader for incoming Gmail messages. The service creates URL addresses for the OAuth protocol. The address itself does not contain any sensitive information about the user or keys. The URL is then sent to the card to be signed and the signature is appended to the address. After the signing operation, content data is retrieved from the service and displayed to the user. When initiating the service for the first time, the user is directed to the services authentication site, where his or her username and password are used to authenticate the user and generate the authentication token used in the card later on. This corresponds to the implementation of the OAuth protocol (Hammer-Lahav, 2010). The OAuth protocol for the smart card is described in Figure 9. In the Java Card 2 prototype, the client application handles the logic for content display and URL creation instead of the applet on the card, since Java Card 2 does not have HTTP connectivity.

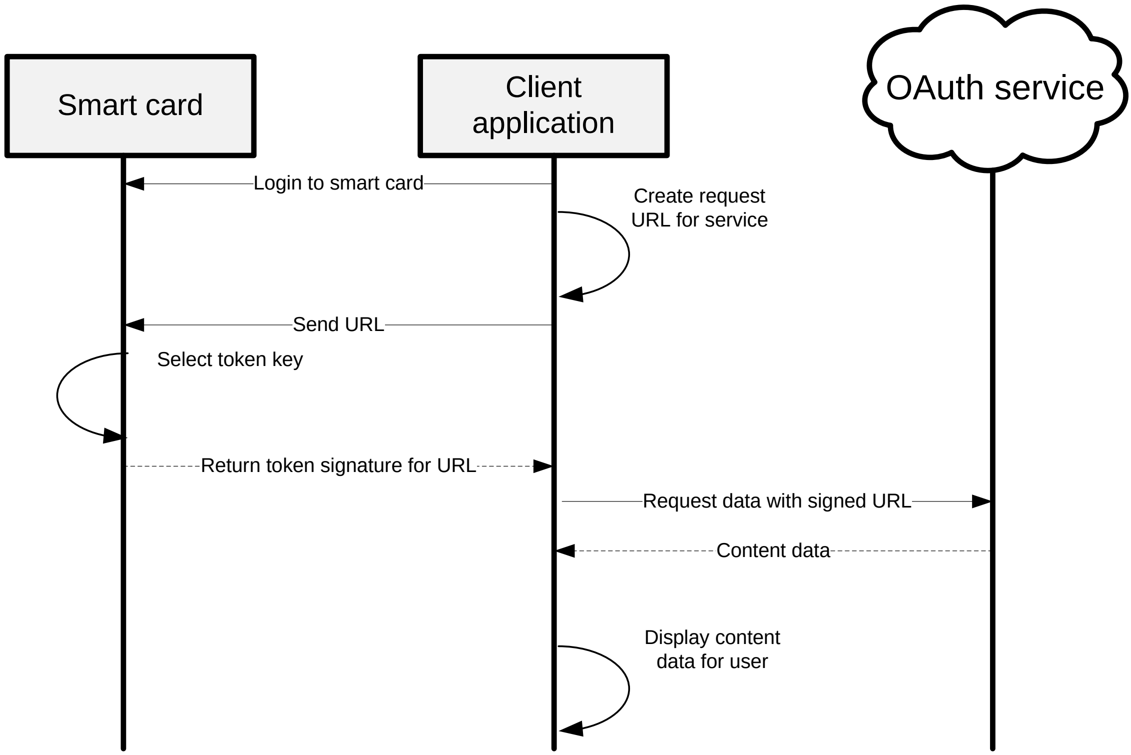

Smart card
Client application
OAuth service
Login to smart card
Create request URL for service
Send URL
Select token key
Return token signature for URL
Request data with signed URL
Content data
Display content data for user

Figure 9. OAuth protocol flow.

Authentication service. The authentication service implements smart card-based session management for websites with two-factor authentication. Identification for a website is done with both a PIN code for the smart card and a token key stored on the smart card. When the user enters a protected web site, a 2D barcode is presented. The user logs in to the prototype application on the phone and scans the barcode. The barcode data is signed on the smart card with the token key and sent to the web service that then authenticates the user and redirects the browser to the authenticated site. Authentication is sent over the 3G or WLAN network from the mobile phone, so no direct connection between the browser and mobile phone is required.

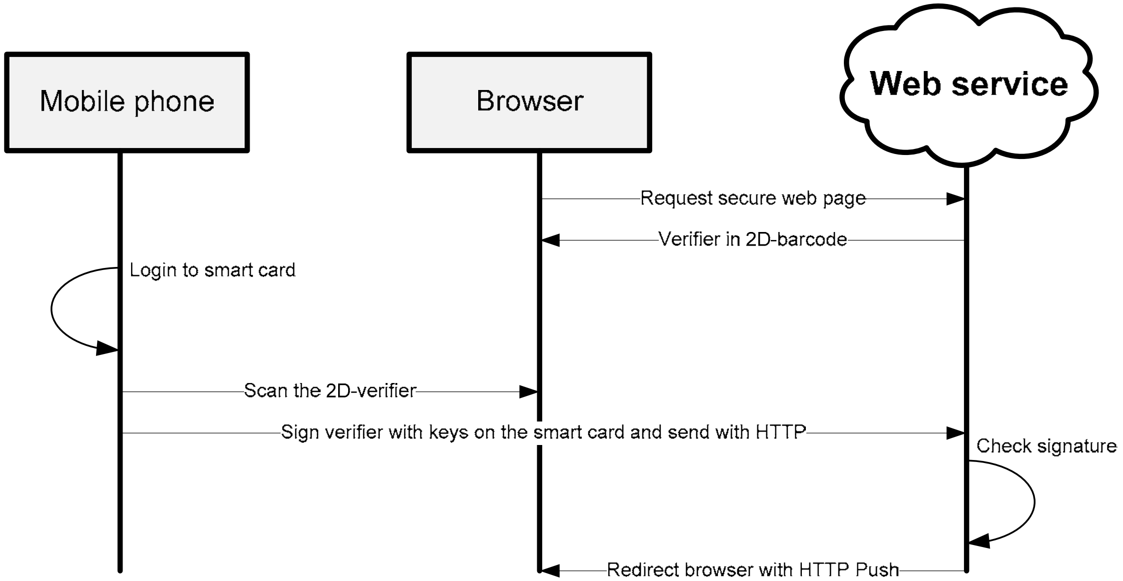


Figure 10. Authentication protocol.

The authentication protocol does not comply with the OAuth protocol, but since it uses the same kind of credentials, the login service can be implemented for multiple services without modifications to the applet.

Ticketing service. In addition to the services described above, a simple ticketing scheme was implemented on the card to provide electronic ticketing for small-scale events with no verifier devices. A ticket is generated by signing a verifier string that is provided by a 2D barcode for the event with a ticket key on the smart card. The verified signature is encoded to a 2D image and displayed at the entrance to the venue. Like the authentication service, ticketing also works with the same credential storage as OAuth without modifications to the applet. The ticketing service is described in greater detail in Appendix A since it is an offline service and not fully in the scope of this work.

5.4.2 Credential model

Figure 11 shows the composition of the credential tree in the prototype. The main difference to the first prototype is the absence of role hierarchy.

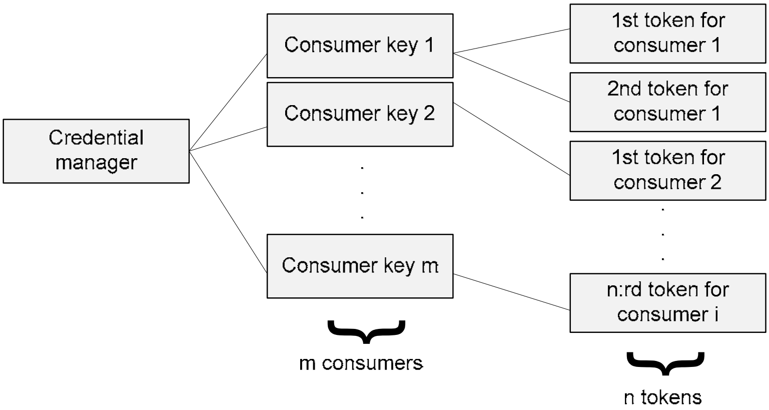


Figure 11. Credential model for the second prototype.

The credential model features OAuth token storage allowing a limited number of different OAuth credentials to be stored on the card. The card applet provides storage and signing functionalities for the client application. All critical data is saved on the card and signed on the card without transferring keys after initialization. HTTP request sending and receiving is handled by the client application, which differs from the first prototype. This is due to platform limitations, as it is not possible to send HTTP calls in Java Card 2. Also, the key policies for the platform are usually very strict, restricting the applet to a very static environment. In order to make the protocol flexible and to support multiple services, an open credential protocol such as OAuth is required. Another possibility is a programmable environment inside a closed secure element similar to the On-board Credentials platform (Ekberg et al., 2008).

5.4.3 Architecture

As shown in Figure 12, in the second prototype much of the functionality is moved from the applet side to the client application, but all the critical data is still processed and stored inside the applet and not revealed after initialization. The client handles all HTTP traffic to the service and the formatting of the data for the user interface. Data is requested with a URL signed on the card side, so no confidential data such as keys are revealed to the client.

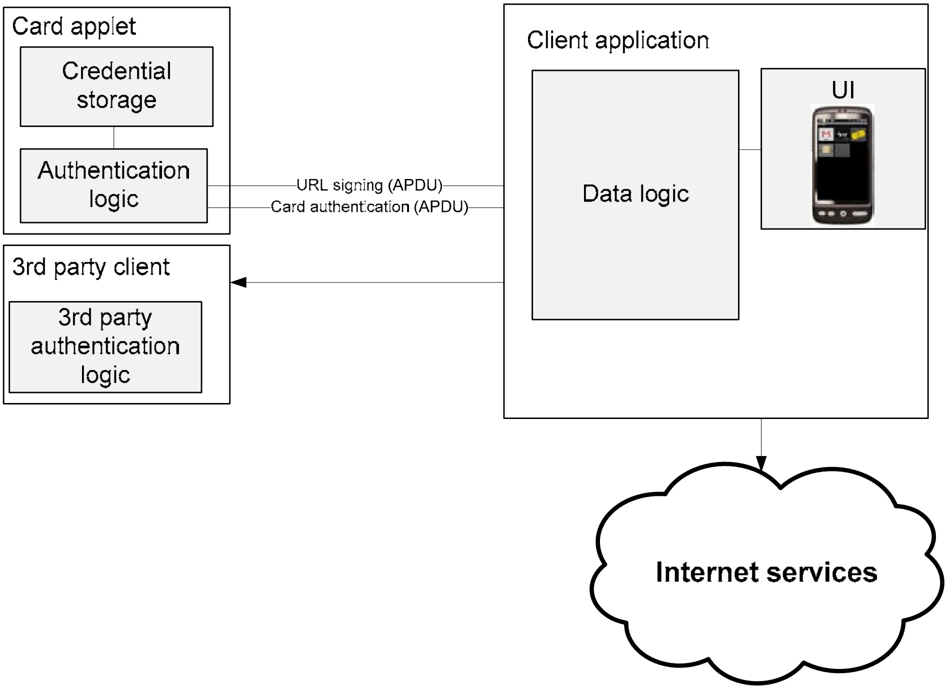


Figure 12. Java Card 2.2 architecture.

With the Java Card 2.2 architecture, the client cannot work as a proxy, since Java Card 2 does not have any web connectivity. The platform is also simplified to support only credential storage and signing compliant with the OAuth protocol. OAuth itself provides a flexible protocol for storing credentials for different services. Credential storage also allows different types of authentication methods to be used if they follow the separate consumer and access token model used in OAuth. Services implemented in this prototype are a Google Gmail reader to demonstrate OAuth functionality, 2D-barcode login and session management for web services demonstrating two-factor authentication with a smart enabled mobile phone. In addition, a verifier-based ticketing demonstration for small events was implemented.

5.4.4 Client application

The Android version of the client application for the Java Card 2 prototype is a minimal version of the application presenting services on the card. When the user starts the program, it requests the PIN code of the card before services can be accessed. After login, all services are accessible with a single login and are displayed in the main menu (Figure 13). The logic for the client application includes forming the content URLs for data retrieval, but signing operations and key management are handled on the card side without the need to expose confidential data on the client side.

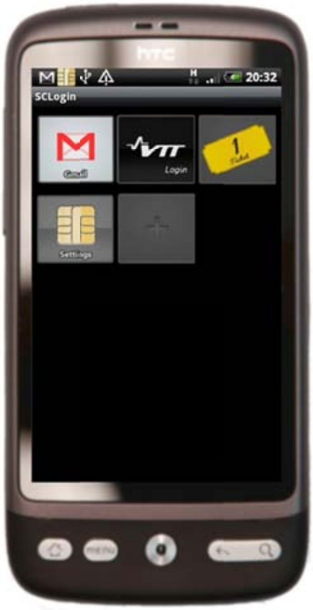


Figure 13. Main menu for Java Card 2 prototype.

The client application is displayed to the user like any other Android application. Usage of secure credential storage is transparent and does not require user interaction except when asking credentials on the card. Connection to the smart card is handled as a background service that can be configured to ask for a PIN code whenever starting the client application or to maintain the connection for a specified amount of time without client usage.

Communication between the client application and the smart card is executed via manufacturer-specific drivers since there is no standardized way for a microSD smart card to communicate with the phone. This differentiates it from a SIM-based smart card, which can communicate with a phone through normal SATSA interfaces with a Java-based client application.

6. Evaluation

The evaluation of the constructed prototype is done based on the requirements derived from the literature, as described in Chapter 4. The evaluation criteria are also compared to the previously constructed Java Card 3 prototype. The viewpoint of the evaluation is in usable security and future directions are identified from the evaluation. The first part of the chapter combines the evaluation criteria and the design choices with the requirements. The main design choice here is the use of single sign-on and an open credential management model. Both the earlier prototype and the prototype in this work are evaluated against the formed criteria although the credential model of the earlier prototype is created before forming the requirements in this work. The main focus of the evaluation is on the prototype constructed in this work and the earlier prototype is used as a reference for upcoming technology.

6.1 Evaluation criteria and design choices combined

The constructed card is evaluated from the viewpoint of usable security. The purpose of the card is to enable an easier way of managing credentials for services and to keep the credentials safe. One approach to achieving this is the creation of a product that is easy to use and open enough to adapt to the user's needs. Hardware security and client software usability are also important aspects, but as the pilot product developed in this work is on a hardware platform that is not in production use, studying these matters is not relevant for the scope of this work.

When comparing the requirements from the literature review introduced in Chapter 4.3 to possible implementation models, there are few alternatives; one of them that fits the requirements is single sign-on.

- **Multiple credentials**

  It was previously established that the credential system must be open and user controllable and that the user must be able to select services and control access delegation. One implementation model is to use a generic authentication framework such as OpenID (Klenk et al., 2009). A generic framework enables the use of multiple different services easily as a multitude of services can be used with the same protocol. This is especially suitable to a smart card environment where access to smart card applets is restricted. The second alternative solution is to create an interactive scripting environment inside a secure element, as demonstrated with the On-board Credentials platform. ObC enables use of third-party authentication scripts inside a protected environment and therefore can support a wide range of different services. (Ekberg et al., 2008).

- **Secure passwords**

  To make the user follow security guidelines, users' cognitive load must be minimized. This can be accomplished by reducing the number of needed passwords and making a password model that fits the users' mental model. Single sign-on is one solution to achieve this as it minimizes the amount of passwords and gives greater control over used passwords.

- **Ease of use**

  Lastly, credential usage must reduce users' cognitive load and be transparent to the user. Single sign-on provides one option for this as it facilitates authentication to the process and makes individual service authentication invisible to the user.

The following table collects the criteria for the evaluation of prototypes. The criteria are compared to the requirements of the problem area using single sign-on as the main design solution.

Table 5. Evaluation criteria.

	Problem area requirements	Criteria
Credential management	Multiple credentials • Open and user controllable • Service access delegation	1. Prototype implements support for open authentication protocol 2. Support for time-based tokens and secure token distribution
	Secure passwords • Minimizing cognitive load • Mental model compatibility	3. Application implements a single sign-on model 4. Enables use of persuasive password strategies 5. Usage is similar to existing software used by user
	Ease of use • Transparent credential usage • Users' knowledge of password protection	6. Automatic single sign-on to services after card authentication 7. Visibility of current protection level
Interoperability	• Platform interoperability • Unified user experience • Synchronization abilities	8. Minimal platform requirements or standardized interface 9. User experience is the same on all platforms used 10. Platform enables secure synchronization and backing up of credentials 11. Minimal client logic

The criteria in Table 5 are used to evaluate both the preceding prototype and the prototype constructed in this work. Both criterion 3 and 6 cover the implementation of single sign-on and are therefore evaluated as one in the following evaluation.

6.2 Java Card 3 prototype

The Java Card 3 prototype was built as a proof of concept using techniques providing the fullest capabilities without accounting for the current technical or political limitations of smart cards available on the market. The prototype was implemented on a Java Card 3 emulator whose memory capabilities surpass those of currently available Java Card 2 generation cards. The purpose of the prototype was to provide a proof of concept of the model with future technologies.

The prototype had OAuth as an open credential model to allow user manageable services. Token delegation is not supported with an applet or directly with OAuth, but the next version of the OAuth protocol supports token expiration and a refresh token that could, at least with the support of the service, provide support for token delegation (Hammer-Lahav, 2011). Support for time-based tokens currently depends on service implementation. The applet also allows other third-party applets to be used, which could enable extendibility beyond the OAuth protocol on the card. Authentication to services is executed with card credentials that can be configured in a more flexible fashion than individual service credentials. It also enables single sign-on to services with one password. This allows use of persuasive strategies to create a long and secure password or few security levels. The prototype in itself does not support key delegation and sharing, but this kind of functionality could be implemented with the support of OAuth 2.0 and the service used (Hammer-Lahav, 2011).

As the card works as an authentication proxy between the client and services, it removes all authentication or data querying logic from the client, leaving only the user interface- and formatting-related logic. This makes the client code easy to port to different platforms. Furthermore, third-party client applications are easy to create. The client could be integrated into existing applications or it could be used as a separate application similar to currently existing software. With separate software, there exists the possibility that users may choose a less secure application if they prefer it more. With integrated software, there is the possibility that users might not know which credentials are protected and when, as is the case with security toolbars (Wu et al., 2006).

Credentials on the card are stored in a very secure fashion. A password is needed to access them. Only the user knows this password and moreover all credentials are stored securely on the card. Without backup capabilities, the loss of a card or card password can cause significant work for the user when all individual service accesses need to be recovered. Back-up abilities and synchronization over multiple cards are out of the scope of this prototype, but there are no technical limitations to implementing such capabilities since all the data can be extracted from the card in encrypted form. Solutions are available that enable third-party access or the recovery of data without compromising security and without the user's password, such as the use of biometric identification and shared secrets (Gardner et al., 2009). This kind of model could be employed to enable password recovery methods controlled by a trusted third party.

The evaluation conducted for this prototype in this chapter is summarized in the following table.

Table 6. Criteria evaluation.

Criteria	**Fulfils**	**Comments**
Open authentication protocol	X	Implemented and user manageable for different service configurations.
Time-based tokens and token delegation	-	Prototype does not support, but could be supported in the next version with OAuth. Support from the service might be required.
Single sign-on model	X	Implemented with a single login password and expandable to different service configurations.
Support for persuasive passwords	X	The use of different types of persuasive strategies is enabled when the number of passwords decreases and control of passwords moves from the service to the applet.
Similarity to existing software	X/?	Client software works exactly like other desktop applications connecting to similar services. However it is not clear in the scope of this work whether users would prefer the prototype application over service web sites.
Protection level visibility	X	All passwords within the application are protected, making the protection level visible.
Minimal platform requirements	X	The card works through normal HTTP interfaces, leading to easy communication with standard methods and interoperability between platforms.
Unified user experience	-	With an HTTP interface, it is easy to port the client application to different platforms. However, an NFC reader would still be required. Currently, readers are not available in every computer, creating limitations as to where the system can be used. If the system cannot be used everywhere, the user still needs to remember individual service passwords.
Synchronization and back-up abilities	-	Not implemented in the prototype, but there are no technical limitations to creating encrypted off-card backups with the available methods. (Gardner et al., 2009).
Minimal client logic	X	The client requires only logic related to the user interface and formatting of data, making it very lightweight. Authentication and service-related logic is done on-card.

6.3 Java Card 2 prototype

The Java Card 2 prototype uses the same credential protocol, but is simplified to better fit the requirements of the platform. This limits some use cases, as support from the service is required for the open authentication protocol. Also, the credential model is simplified by only using service tokens without role level. Because the credential protocol implementation is the same as in the previous prototype, all the same properties and constraints apply to it. It implements an extendable protocol with a single sign-on. Expiration depends on the service and token delegation could be supported in the next version of OAuth. (Hammer-Lahav, 2011).

As with the Java Card 3 prototype, this prototype also enables support for persuasive password strategies. Only one password is required to access on-card credentials, which allows the user more freedom to use long passphrases or graphical passwords enabled by the client.

The platform for the prototype is a mobile device on which the secure element is installed and available continuously. On the other hand, the secure element is not easily movable to another device, weakening availability on multiple devices.

Table 7. Criteria evaluation.

Criteria	**Fulfils**	**Comments**
Open authentication protocol	X	OAuth implemented and extendable to multiple services.
Time-based tokens and token delegation	-	Prototype does not support, but could be supported in the next version with OAuth. Support from the service might be required.
Single sign-on model	X	Implemented with a single login password and expandable to different service configurations.
Support for persuasive passwords	X	The use of different types of persuasive strategies is enabled when the number of passwords decreases and control of passwords moves from the service to the applet.
Similarity to existing software	X	In a mobile environment services are frequently available in separate applications besides the browser application. The use paradigm of the prototype is similar to these applications.
Protection level visibility	X	Credentials are always protected inside the prototype application.

Minimal platform requirements	X/?	A microSD slot is available in most phone models, but to use it, special drivers are required and APDU-based communication is used. In the future, a standardized connectivity framework, such as the Mobile Open API proposed by SIMAlliance, could provide easier communication with the card. (SIMAlliance, 2011).
Unified user experience	-	The application itself is portable to different platforms. However, a microSD is not easily removable from all phone models, and even if an NFC phone would be used, not every computer has a reader. This limits where the system can be used and users would still be required to remember service passwords.
Synchronization and back up	-	Not implemented in the prototype, but there are no technical limitations to creating encrypted off-card backups with the available methods. (Gardner et al., 2009).
Minimal client logic	-	The client is required to perform at least some parts of forming authentication URLs and the actual data querying. This limitation is due to the limited capabilities of the Java Card 2 platform. At least some client logic is required besides user interface formatting in comparison to the Java Card 3 prototype.

The credential applet fulfils the requirements to create a usable credential applet and the open credential protocol creates a secure platform for a more usable application requiring authentication. The evaluation and the need for further research on the client application are discussed in greater depth in the next chapter.

The credential model reduces the number of passwords, effectively transfers sensitive information from the application to the secure element and enables the use of single sign-on and persuasive password strategies. As no authentication-related information is transferred out from the card when using the authentication protocol, it provides a very secure environment for credentials and authentication. Authentication on the card does not depend on the services or service giving control of the password to a credential manager. This enables the use of user-centric password mechanisms such as graphical passwords. However, when password management is integrated as a single sign-on-based solution with only a single master password, all the passwords could also be vulnerable to the password domino effect when reusing passwords (Ives et al., 2004). It is possible to integrate different security levels, for instance, to better protect access to different services. In some cases, biometric authentication methods could be used. Also, since the credential manager only stores access tokens instead of actual passwords, they can be revoked without changing passwords.

Although a deeper evaluation of the client is out of the scope of this work, the constructed prototype client has some problems similar to those of existing password managers. The visibility of the protection level is clear when using the prototype client, since all the credentials within the software are protected. However, if the application would be integrated more tightly into the existing architecture, the visibility of protection could suffer. The credential manager still requires support from the service in the form of an open credential protocol and, as such, not all existing services can be bound to the credential manager. Because of this, it is not certain whether the protection level would be clear and visible to the user if the client were integrated tightly with an existing mobile platform. Further research on the client application with an existing credential model is required to better identify usability requirements for the client. These are also technical problems with currently existing password management tools (Chiasson et al., 2006; Whitten & Tygar, 1999).

From the hardware perspective, the requirements are implementable on most Java Card platforms, as the only requirement is the availability of correct signing algorithms. The prototype constructed in this work uses a microSD smart card that was tested on two different card platforms to achieve hardware consistency. Although card operating systems are similar to other Java Card platforms, interfaces between the card and the device are not standardized. Special driver software is usually required, reducing the availability of credential storage on multiple devices. Similarly, some client logic related to service authentication and data retrieval is required with Java Card 2, reducing interoperability between different platforms. Key management with microSD-based smart cards is similar to SIM-based cards or an embedded secure element since they all usually use standard Java Card operating systems. Key management policies are still more relaxed compared to other mobile phone secure elements since ownership of microSD cards is not controlled by operators or phone manufacturers. Upcoming standardization efforts such as SIMAlliance’s Open Mobile API and Java Card 3 could bring improvements to interoperability by introducing a more standardized interface to access different or multiple secure elements (SIMAlliance, 2011; Sun Microsystems, 2008).

7. Conclusion

The main research question set for this study was to identify whether a secure element with web-connected services can make credential management easier and more secure from the viewpoint of usable security. It was further divided into three parts: Identity and credential management strategies, requirements for a secure element-based credential manager and the implementation of such a credential manager applet prototype.

The major part of the study concerned identifying the requirements for the credential manager. In the study, a comprehensive literature review was conducted to ascertain the credential manager requirements and provide an overview of user password management strategies. As it was executed in a formal way, the review provides a solid basis for the requirements. It considered key aspects for enabling users to create stronger passwords. It was found that users consider security policies to be a burden and also that having to remember a great number of passwords creates too much cognitive load for them. Furthermore, the security requirements might not be implemented in line with the users' mental model, creating unrealistic expectations for security. The literature review also revealed different types of solutions for credential platform implementation, but many of these solutions implement a credential model with some degree of openness when multiple credentials are required. Some even define a model for access delegation or different access groups. These findings indicated that the current password management strategies of users correspond poorly to actual security requirements, thereby creating very real security threats. Findings for credential design indicated that there is also a strong tendency towards a more open credential model when using multiple services with a credential manager. This model is considerably different from the current model of web services, where control of passwords is handled by individual services.

Requirements for the credential manager were identified from the literature and a credential manager model was designed accordingly. The study identified a feasible model to be used with secure elements to achieve support for single sign-on-based credential management. Requirements relating to the usability of the model from the users' point of view were also identified, providing one model for further research on the subject.

Based on the findings of the literature review, an open credential model was chosen in order to reduce the number of used passwords and to create transparent single sign-on authentication. These findings were also used to form the evaluation criteria used in this work. The prototype application was built to test the credential model and evaluate it. In addition, the prototype built prior to this work was used to provide additional perspective and to test evaluation criteria. Both prototypes fulfilled the requirements in the most critical respects in terms of open authentication and the credential model. Moreover, they are met with minimal platform requirements, requiring only basic credential storage and signing at minimum. The prototypes show that a secure element-based credential manager that can be used to support web services with open credentials is implementable on the target platform.

Further analysis showed that the credential model itself effectively reduced the number of passwords needed to use services, and also enabled the use of more flexible and transparent on-card authentication models, with promising results. On the other hand, the analysis could not answer whether the client application can provide a unified user experience and visibility of protection level, leaving a need for further research. Also, further research is required on the credential protocol and the credential model itself to provide more user-centric features, such as time-based token delegation and synchronization abilities.

From the hardware point of view this study showed that the current secure element implementation provides a platform that can support the credential model itself with only minimal requirements. Implementation details are similar to those of both tested cards and they are also very similar to other platforms such as SIM cards or embedded secure elements. However, the interface between the secure element and client application is not very standardized yet, although there are upcoming efforts such as SIMAlliance's Open Mobile API that could provide a more standardized way to access the credential manager (SIMAlliance, 2011). Also, the key management policies of current cards can pose difficulties to the integration of the credential manager into existing secure element platforms. Both of these issues raise the need for further research.

References

Adams, A. & Sasse, M. A. 1999. Users are not the enemy. *Communications of the ACM,* 42(12), 46.

Adams, A. & Sasse, M. A. 2001. Privacy in multimedia communications: Protecting users, not just data. *Joint Proceedings of HCI 2001 and IHM 2001: People and Computers XV: Interactions without Frontiers,* 49.

Akram, R., Markantonakis, K. & Mayes, K. 2009. Application management framework in user centric smart card ownership model. *Information Security Applications,* 5932(2009), 20–35.

Alzomai, M. & Jøsang, A. 2010. The mobile phone as a multi OTP device using trusted computing. *Proceedings of 4th International Conference on Network and System Security (NSS),* pp. 75–82.

ARM 2009. *ARM security technology – building a secure system using TrustZone® technology.* Retrieved 1.6.2011, from http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf (1.9.2011).

Barnes, S. B. 2006. A privacy paradox: Social networking in the United States. *First Monday,* 11(9).

Bauer, L., Cranor, L. F., Reiter, M. K. & Vaniea, K. 2007. Lessons learned from the deployment of a smartphone-based access-control system. *Proceedings of the 3rd Symposium on Usable Privacy and Security,* 75.

Bichsel, P., Camenisch, J., Groß, T. & Shoup, V. 2009. Anonymous credentials on a standard java card. *Proceedings of the 16th ACM Conference on Computer and Communications Security,* 600–610.

Brickell, E. & Li, J. 2007. Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities. *Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society,* 30.

Brodie, C., Karat, C. M., Karat, J. & Feng, J. 2005. Usable security and privacy: A case study of developing privacy management tools. *Proceedings of the 2005 Symposium on Usable Privacy and Security,* 43.

Chen, Z. 2000. *Java card technology for smart cards: Architecture and programmer's guide.* Prentice Hall PTR.

Chiasson, S., Forget, A., Biddle, R. & van Oorschot, P. C. 2008. Influencing users towards better passwords: Persuasive cued click-points. *Proceedings of the 22nd British HCI Group Annual Conference on HCI 2008: People and Computers XXII: Culture, Creativity, Interaction-Volume* 1*,* 121–130.

Chiasson, S., Van Oorschot, P. C. & Biddle, R. 2006. A usability study and critique of two password managers. *Proceedings of the 15th USENIX Security Symposium,* 1–16.

Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D. & Mitchell, J. C. 2004. Client-side defense against web-based identity theft. *Proceedings of 11th Annual Network and Distributed System Security Symposium (NDSS '04).*

Cranor, L. F. & Garfinkel, S. 2005. *Security and usability: Designing secure systems that people can use.* O'Reilly Media, Inc.

Ekberg, J. E., Asokan, N., Kostiainen, K. & Rantala, A. 2008. On-board credentials with open provisioning. *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security,* 104–115.

Emigh, A. 2005. Online identity theft: Phishing technology, chokepoints and countermeasures: ITTC Report on Online Identity Theft Technology and Countermeasures.

Farrel, S. 2004. *RFC3767: Securely available credentials protocol.*

Fogg, B. J. 2002. Persuasive technology: Using computers to change what we think and do. *Ubiquity, 2002*(December), 2.

Forget, A., Chiasson, S., van Oorschot, P. & Biddle, R. 2008. Persuasion for stronger passwords: Motivation and pilot study. *Proceedings of the 3rd International Conference on Persuasive Technology,* 140–150.

Gajek, S., Sadeghi, A. R., Stuble, C. & Winandy, M. 2007. Compartmented security for browsers – or how to thwart a phisher with trusted computing. *Proceedings of the Second International Conference on Availability, Reliability and Security, 2007. ARES 2007,* 120–127.

Gardner, R. W., Garera, S., Pagano, M. W., Green, M. & Rubin, A. D. 2009. Securing medical records on smart phones. *Proceedings of the First ACM Workshop on Security and Privacy in Medical and Home-Care Systems,* 31–40.

Gaw, S. & Felten, E. W. 2006. Password management strategies for online accounts. *Proceedings of the Second Symposium on Usable Privacy and Security,* 55.

GlobalPlatform 2010. *GlobalPlatform – technical overview.* Retrieved 16.8.2010, from http://www.globalplatform.org/specifications.asp

Gross, J. B. & Rosson, M. B. 2007. Looking for trouble: Understanding end-user security management. *Proceedings of the 2007 Symposium on Computer Human Interaction for the Management of Information Technology,* 30–31.

Gross, R., Acquisti, A. & Heinz, H. J. 2005. Information revelation and privacy in online social networks. *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society,* 80.

Gühring, P. 2007. *Concepts against man-in-the-browser attacks.* Retrieved 23.6.2011, from http://www.cacert.at/svn/sourcerer/CAcert/SecureClient.pdf

Hammer-Lahav, E. 2010. *The OAuth 1.0 protocol.* Retrieved 1.6.2011, from http://tools.ietf.org/html/rfc5849

Hammer-Lahav, E. 2011. *The OAuth 2.0 authorization protocol, draft-ietf-oauth-v2-15.* Retrieved 7.4.2011, from http://tools.ietf.org/html/draft-ietf-oauth-v2-15

Herley, C. 2009. So long, and no thanks for the externalities: The rational rejection of security advice by users. *Proceedings of the 2009 Workshop on New Security Paradigms Workshop,* 133–144.

Herzberg, A. & Gbara, A. 2004. *Trustbar: Protecting (even naive) web users from spoofing and phishing attacks.* Computer Science Department, Bar-Ilan University.

Hevner, A. R., March, S. T., Park, J. & Ram, S. 2004. Design science in information systems research. *Mis Quarterly,* 28(1), 75–105.

Horcher, A. M. & Tejay, G. P. 2009. Building a better password: The role of cognitive load in information security training. *Proceedings of IEEE International Conference on Intelligence and Security Informatics, 2009. ISI'09,* 113–118.

Husemann, D. 1999. The smart card: Don't leave home without it. *IEEE Concurrency, 7*(2), 24–27.

Hyppönen, K. 2008. An open mobile identity tool: An architecture for mobile identity management. *Public Key Infrastructure,* 207–222.

Inglesant, P. G. & Sasse, M. A. 2010. The true cost of unusable password policies: Password use in the wild. *Proceedings of the 28th International Conference on Human Factors in Computing Systems,* 383–392.

Ives, B., Walsh, K. R. & Schneider, H. 2004. The domino effect of password reuse. *Communications of the ACM,* 47(4), 75–78.

Jøsang, A. & Pope, S. 2005. User centric identity management. *Proceedings of AusCERT Asia Pacific Information Technology Security Conference.*

Kalofonos, D. N. & Shakhshir, S. 2007. IntuiSec: A framework for intuitive user interaction with smart home security using mobile devices. *Proceedings of International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th,* 1–5.

Keith, M., Shao, B. & Steinbart, P. J. 2007. The usability of passphrases for authentication: An empirical field study. *International Journal of Human-Computer Studies,* 65(1), 17–28.

Kelton, K., Fleischmann, K. R. & Wallace, W. A. 2007. Trust in digital information. *Journal of the American Society for Information Science and Technology,* 59(3), 363–374.

Khan, S. & Hayat, A. 2009. A trustworthy identity management architecture for e-government processes. *Proceedings of World Congress on Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS'09.* 231–234.

Kitchenham, B. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University,* 33.

Klenk, A., Kinkelin, H., Eunicke, C. & Carle, G. 2009. Preventing identity theft with electronic identity cards and the trusted platform module. *Proceedings of the Second European Workshop on System Security,* 44–51.

Li, Q., Zhang, X., Seifert, J. P. & Zhong, H. 2008. Secure mobile payment via trusted computing. *Proceedings of Trusted Infrastructure Technologies Conference, 2008. APTC'08. Third Asia-Pacific,* 98–112.

Madlmayr, G. 2008. A mobile trusted computing architecture for a near field communication ecosystem. *Proceedings of the 10th International Conference on Information Integration and Web-Based Applications & Services,* 563–566.

Maher, D. P. 1998. A platform for privately defined currencies, loyalty credits, and play money. *Financial Cryptography,* 1465/1998, 43.

Mannan, M. & van Oorschot, P. 2008. Security and usability: The gap in real-world online banking. *Proceedings of the 2007 Workshop on New Security Paradigms,* 1–14.

Nissenbaum, H. 1998. Protecting privacy in an information age: The problem of privacy in public. *Law and Philosophy,* 17(5), 559–596.

Nunamaker, J. F., Chen, M. & Purdin, T. D. M. 1990. Systems development in information systems research. *Journal of Management Information Systems,* 7(3), 89–106.

Open Mobile Alliance 2008. *Enabler release definition for smartcard-web-server.* Retrieved 1.6.2011, from http://www.openmobilealliance.org/technical/release_program/docs/SCWS/V1_0-20080421-A/OMA-ERELD-Smartcard_Web_Server-V1_0-20080421-A.pdf

OpenID Foundation 2010. *OpenID and OAuth hybrid extension.* Retrieved 20.3.2011, from http://wiki.openid.net/w/page/12995194/OpenID-and-OAuth-Hybrid-Extension

Recordon, D. & Reed, D. 2006. OpenID 2.0: A platform for user-centric identity management. *Proceedings of the Second ACM Workshop on Digital Identity Management,* 16.

Sandhu, R. & Zhang, X. 2005. Peer-to-peer access control architecture using trusted computing technology. *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies,* 158.

SIMAlliance 2011. *Open mobile API: An introduction.* Retrieved 1.6.2011, from http://www.simalliance.org

Simon, H. A. 1996. *The sciences of the artificial.* The MIT Press.

Singh, S., Cabraal, A., Demosthenous, C., Astbrink, G. & Furlong, M. 2007. Security design based on social and cultural practice: Sharing of passwords. *Usability and Internationalization.Global and Local User Interfaces,* 476–485.

Singh, S., Cabraal, A. & Hermansson, G. 2006. What is your husband's name?: Sociological dimensions of internet banking authentication. *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments,* 237–244.

Sun Microsystems 2008. *The java card™ 3 platform white paper.* Retrieved 1.6.2011, from The Java Card™ 3 Platform white paper.

Sun Microsystems 2009. *Java™ servlet specification, java card™ platform, version 3.0.1.* Retrieved 1.6.2011, from http://java.sun.com/javacard/3.0.1/specs.jsp

Toegl, R. & Hutter, M. 2010 An approach to introducing locality in remote attestation using near field communications. *The Journal of Supercomputing,* 1–21.

Trusted Computing Group 2003. *TCG specification architecture overview.* Retrieved 1.6.2011, from http://www.trustedcomputinggroup.org/files/resource_files/AC652DE1-1D09-3519-ADA026A0C05CFAC2/TCG_1_4_Architecture_Overview.pdf

Vandewalle, J. J. 2005. Smart card research perspectives. *Construction and Analysis of Safe, Secure, and Interoperable Smart Devices,* 3362(2005), 250–256.

Weirich, D. & Sasse, M. A. 2001. Pretty good persuasion: A first step towards effective password security in the real world. *Proceedings of the 2001 Workshop on New Security Paradigms,* 143.

West, R. 2008. The psychology of security. *Communications of the ACM,* 51(4), 34–40.

Whitten, A. & Tygar, J. D. 1999. Why johnny can't encrypt: A usability evaluation of PGP 5.0. *Proceedings of the 8th USENIX Security Symposium,* 169–184.

Wu, M., Miller, R. C. & Garfinkel, S. L. 2006. Do security toolbars actually prevent phishing attacks? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 610.

Yan, J. 2004. Password memorability and security: Empirical results. *Security & Privacy, IEEE,* 2(5), 25–31.

Appendix A: 2D ticketing service

During this work, the need for a small event ticketing prototype emerged. The existing credential platform was used to create a fast prototype for a ticketing application. The ticketing service was intended as a low-cost electronic ticketing service for small-scale events where online ticketing is not available. Ticket verification was done entirely by using a 2D verifier on the phone. This makes the ticket less secure than offline and online tickets verified with methods such as NFC. The advantage is that the ticket requires only a smart phone with a camera and secure element. Offline ticketing is in no sense compatible with OAuth, which is intended for service authentication. It is however possible to use the same credential storage and signing functions to provide ticketing service without modifications to the applet.

When the ticketing application is installed, the application key is installed on the card with an encrypted message. The application key corresponds to the consumer key used with the OAuth protocol. Individual tickets purchased are stored as token keys. When the user enters the event venue, the verifier string is presented in the form of a 2D barcode. The user scans the barcode with a phone camera and it is signed on the card with a consumer and token key combination. The signature is used to generate a 2D image that is then presented at the entrance to the venue and verified visually. The complete ticketing process is presented in the following figure.

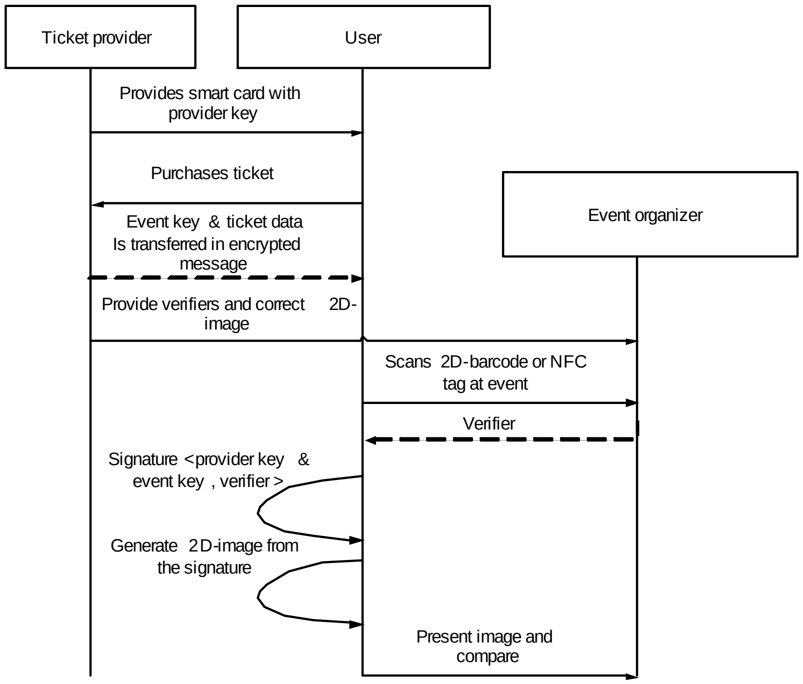

Ticket provider
User
Event organizer
Provides smart card with provider key
Purchases ticket
Event key & ticket data Is transferred in encrypted message
Provide verifiers and correct 2D-image
Scans 2D-barcode or NFC tag at event
Verifier
Signature <provider key & event key , verifier >
Generate 2D-image from the signature
Present image and compare

Figure 14. Ticketing process.

The generated ticket is not secure after verification since the 2D image is relatively easy to duplicate. The advantage is that verification does not require any special devices and the requirements for the users' phones can be minimal; only the secure element is needed, and it can be delivered on a secure memory card or with a SIM card. The protocol presents one alternative to electronic ticketing cases, where the costs have to be kept low and monetary losses in case of duplication are minimal.

Series title, number and
report code of publication

VTT Research Notes 2596
VTT-TIED-2596

Author(s)
Aki-Petteri Leinonen

Title

Identity management for web-enabled smart card platform

Abstract

The amount of sensitive information stored in different online services is rapidly growing in traditional web applications and also in mobile services. Most of these services control their own authentication credentials, increasing the number of credentials that the user needs to manage. Previous literature shows that when the amount of passwords grows, users tend to create weaker passwords or reuse passwords for different services. This exposes users to security threats. Attacks target weak passwords and compromised security might result in a domino effect, with one exposed password giving access to multiple services. In addition to usability issues, the mobile platform is vulnerable to physical attacks if the device is lost or stolen. This creates a need for a secure credential management platform for mobile devices that addresses these problems and creates a usable environment for the management of credentials. One such solution could be provided by a secure element inside the mobile device. The secure element is special hardware that provides secure code execution for the mobile platform, as in existing smart card platforms.

This study is based on work carried out at VTT Technical Research Centre of Finland, in which a prototype version of a single sign-on Java Card application was created. The purpose of this study is to find out whether a secure element with support for web-connected services can be used to provide a user-centric credential management platform in a mobile phone. This main question can be divided into three questions that need to be answered: What are the users' password management strategies, what requirements can be identified for a user-centric credential manager inside a secure element and can this solution be implemented with existing technology?

In order to find out password management strategies and existing implementations, an extensive literature review is conducted. With respect to the use of password management strategies, the literature review indicated that users circumvent security methods because they consider that these methods take too much of their time and other resources compared to the perceived gains. It was also revealed that the authentication procedure must follow the user's mental model and not restrict the primary task the user needs to achieve. The requirements for the credential model are identified from the literature review and a single sign-on protocol is chosen to be the approach in this work. A prototype application that allows users to authenticate to different services with a single sign-on, and which also demonstrates two-factor authentication, is built and evaluated along with the earlier prototype. The prototype application built in this work shows that the credential manager application can be implemented in an open manner with a single sign-on protocol.

ISBN
978-951-38-7780-4 (soft back ed.)
978-951-38-7781-1 (URL: http://www.vtt.fi/publications/index.jsp)

Series title and ISSN	Project number
VTT Tiedotteita – Research Notes 1235-0605 (soft back ed.) 1455-0865 (URL: http://www.vtt.fi/publications/index.jsp)	74237

Date	Language	Pages
September 2011	English	64 p. + app. 2 p.

Name of project	Commissioned by
Smart urban spaces	

Keywords	Publisher
Identity management, smart card, mobile web services, single sign-on	VTT Technical Research Centre of Finland P.O. Box 1000, FI-02044 VTT, Finland Phone internat. +358 20 722 4520 Fax +358 20 722 4374

VTT Tiedotteita – Research Notes

2582 Hannu Hänninen, Anssi Brederholm, Tapio Saukkonen, Mykola Evanchenko, Aki Toivonen, Wade Karlsen, Ulla Ehrnstén & Pertti Aaltonen. Environment-assisted cracking and hot cracking susceptibility of nickel-base alloy weld metals. 2011. 152 p.

2583 Jarmo Alanen, Iiro Vidberg, Heikki Nikula, Nikolaos Papakonstantinou, Teppo Pirttioja & Seppo Sierla. Engineering Data Model for Machine Automation 2011. 131 p.

2584 Maija Ruska & Juha Kiviluoma. Renewable electricity in Europe. Current state, drivers, and scenarios for 2020. 2011. 72 p.

2585 Paul Buhanist, Laura Hakala, Erkki Haramo, Katri Kallio, Kristiina Kantola, Tuukka Kostamo & Heli Talja. Tietojärjestelmä osaamisen johtamisessa – visiot ja käytäntö. 2011. 36 s.

2586 Elina Rusko, Sanna Heiniö, Virpi Korhonen, Jali Heilmann, Toni-Matti Karjalainen, Panu Lahtinen & Marja Pitkänen. Messenger Package – Integrating Technology, Design and Marketing for Future Package Communication. Final Report. 2011. 90 p.

2587 Markus Olin, Kari Rasilainen, Aku Itälä, Veli-Matti Pulkkanen, Michal Matusewicz, Merja Tanhua-Tyrkkö, Arto Muurinen, Lasse Ahonen, Markku Kataja, Pekka Kekäläinen, Antti Niemistö, Mika Laitinen & Janne Martikainen. Bentoniittipuskurin kytketty käyttäytyminen. Puskuri-hankkeen tuloksia. 2011. 86 s.

2588 Häkkinen, Kai. Alihankintayhteistyön johtamisesta metalliteollisuudessa. 2011. 71 s.

2589 Pasi Ahonen. Constructing network security monitoring systems (MOVERTI Deliverable V9). 2011. 52 p.

2590 Maija Ruska & Lassi Similä. Electricity markets in Europe. Business environment for Smart Grids. 2011. 70 p.

2591 Markus Jähi. Vartiointipalvelujen arvonmuodostus asiakkaan näkökulmasta. 2011. 91 s. + liitt. 6 s.

2592 Jari M. Ahola, Jani Hovila, Eero Karhunen, Kalervo Nevala, Timo Schäfer & Tom Nevala. Moni-teknisen piensarjatuotteen digitaalinen tuoteprosessi. 2011. 121 s. + liitt. 37 s.

2593 Mika Nieminen, Ville Valovirta & Antti Pelkonen. Systeemiset innovaatiot ja sosiotekninen muutos. Kirjallisuuskatsaus. 2011. 80 s.

2595 Martti Flyktman, Janne Kärki, Markus Hurskainen, Satu Helynen & Kai Sipilä. Kivihiilen korvaaminen biomassoilla yhteistuotannon pölypolttokattiloissa. 2011. 65 s. + liitt. 33 s.

2596 Aki-Petteri Leinonen. Identity management for web-enabled smart card platform. 2011. 64 p. + app. 2 p.

ISBN 978-951-38-7780-4 (soft back ed.)
ISSN 1235-0605 (soft back ed.)

ISBN 978-951-38-7781-1 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1455-0865 (URL: http://www.vtt.fi/publications/index.jsp)

• • • VTT RESEARCH NOTES 2596 IDENTITY MANAGEMENT FOR WEB-ENABLED SMART CARD PLATFORM