Recovering from a Lost Digital Wallet

Raja Naeem Akram
Cyber Security Lab, Department of Computer Science,
University of Waikato.
Waikato, New Zealand
Email: rnakram@waikato.ac.nz

Konstantinos Markantonakis and Keith Mayes
Information Security Group, Smart Card Centre,
Royal Holloway, University of London.
Egham, United Kingdom
Email: {K.Markantonakis, Keith.Mayes}@rhul.ac.uk

***Abstract*—Multi-application smart cards enable a user to have multiple applications on her smart card. The growing trend of services convergence fuelled by the Near Field Communication and smartphones has made multi-application smart cards a tangible reality. In such an environment, cardholders might have number of applications on their smart cards and in case they lose the smart card, they would lose all of the applications. Currently, the recovery of a smart card based service might take from a day to a week at best, during which time the service provider might lose on business from the user because she is not able to access the respective services. The proposed framework in this paper enables a user to acquire a new smart card as she desires and then migrate/restore all of her applications onto it — facilitating her to recover from her lost digital wallet in a secure, efficient, seamless and ubiquitous manner.**

***Keywords*-Smart Card, GlobalPlatfrom Consumer-Centric Model, Java Card, Trusted Computing, Performance Measurement**

I. Introduction

The smart card technology has the capability to have multiple applications coexisting on a single smart card chip in a secure and reliable manner [1]. This initiative is generally termed as multi-application smart cards.

In recent years the convergence of multiple services onto a single smart card has gain moment due to an emergence of Near Field Communication (NFC) [2]. The NFC enables a mobile phone to emulate a contactless smart card. Therefore, a user can use her mobile phone to gain access to different services (i.e. banking, transport and door access etc.). The GSM [3] and GlobalPlatform [4] specifications are also evolved to support the convergence of multiple services in the Issuer Centric Model [5] by including an entity termed as Trusted Service Manager (TSM) [6]. The TSM is a neutral third party that has the administrative control of the smart card. The administrative control includes the installation and deletion of an application from their (issued) smart cards.

In contrast the User Centric Smart Card Ownership Model (UCOM) delegates the ownership of the smart card to its users [5]. The term ownership means the privilege to install or delete an application according to the smart card user's requirements. In such a dynamic and open environment where users can have multiple applications of their choice also create certain security and privacy issues [5], [7]. In March 2012, GlobalPlatform announced the initiative of a user centric ownership model for smart cards termed as GlobalPlatform Consumer-Centric Model [8]. This model is significantly similar to the UCOM; therefore, the proposal in this paper also applies to the GlobalPlatform Consumer-Centric Model.

One of the main features of the UCOM is dynamism (wherever, whenever). This increases the potential damage if the device is lost. To expedite the recovery process after theft or loss, customers should be able to have their applications backed up and then restored when required in a secure and ubiquitous manner to their new devices.

A. Contribution

A backup mechanism enables a user to backup her smart card contents. In adverse circumstances, such as losing her smart card, she could retrieve and restore the contents onto a new smart card from the backup. Furthermore, a similar mechanism referred to as a migration mechanism can also be used if a user decides to upgrade to a new feature-rich smart card. In this paper, we propose backup and migration mechanisms along with associated challenges.

There are some subtle challenges to backup and migration mechanisms in the UCOM, especially in the case of card-bound application leases[1] that restrict applications to their host smart cards. There is also a possibility that the remote location (e.g. backup server) might not be tamper-resistant and a malicious user could take advantage of it. Therefore, it would be safe to assume that instead of transferring whole applications (i.e. code and data), we should only transfer application download credentials. These credentials can be considered as authorisation tokens that are issued by the respective Service Providers (SPs), so a user could use them to acquire the respective application in future.

B. Paper Structure

Section II briefly describes the UCOM for completeness, so the discussion in the paper can be self-contained along with the motivation for the proposed framework. Subsequently, in section III provides a description of the proposed backup and migration framework for smart cards. In section IV, we

[1]Card Bound Application Lease: In this lease, a Service Provider (SP) issues its application to a specific smart card and that instance of the lease is bound to the smart card [9]. In this scenario, the SP will only issue one lease per user, which she can have on any of her smart cards; examples of such a lease may be credit card and (U)SIM card applications.

discuss the implementation and performance measurement of the proposed framework on Java Cards. Section V, analyse the proposed backup and migration mechanisms. Finally, section VI provides the conclusion of the paper.

II. User Centric Smart Card Ownership Model

In this section we begin the discussion with a short introduction to the User Centric Smart Card Ownership Model (UCOM) so the discussion in the paper would be better followed without requiring any additional readings (referenced work).

A. Brief Introduction

In the Issuer Centric Smart Card Ownership Model (ICOM), the organisations that issues smart cards to their customers have the ownership of the smart card as shown in figure 1. They control the contents and functionality supported by the smart card [10]. However, in UCOM the user has the "freedom of choice" in terms of content and functionality on their smart cards. With content we mean the applications that a user could install or delete from her smart card. The functionality concerns with the capability of a smart card like memory space, computational power, security certifications and supported cryptographic algorithms etc.

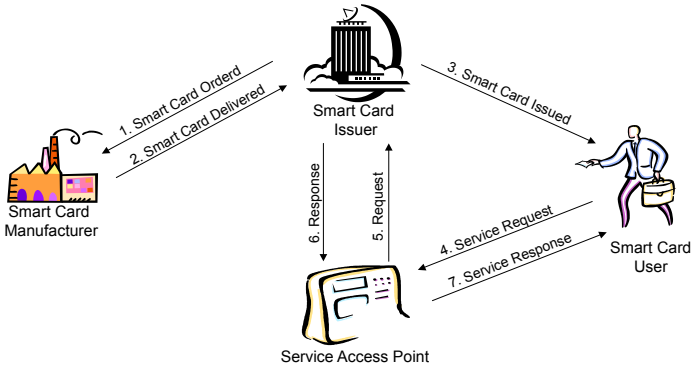

Figure 1. Issuer Centric Smart Card Ownership Model

In the UCOM framework (shown in figure 1), a user acquires a UCOM supported smart card from a card manufacturer. The card manufacturer provides a capability list of the smart card and the user chooses the card that best suites her requirement. At this point the smart card might be a blank card and it does not have any ownership credentials. The user would first generate the ownership credentials and once the user has the taken the ownership of the smart card it could then request a Service Provider (SP) to lease its application(s). An SP is an organisation that develops smart card application(s) and makes them available to their registered customers to download onto their smart cards.

Applications are downloaded to smart cards under the terms and conditions of their respective SPs that are stipulated in the Application Lease Policy (ALP) [9]. If the smart card satisfies the stated ALP, the SP will lease their application to the user's smart card. After downloading the application, the user could present their smart card at a services access point to access the services provided by the SP.

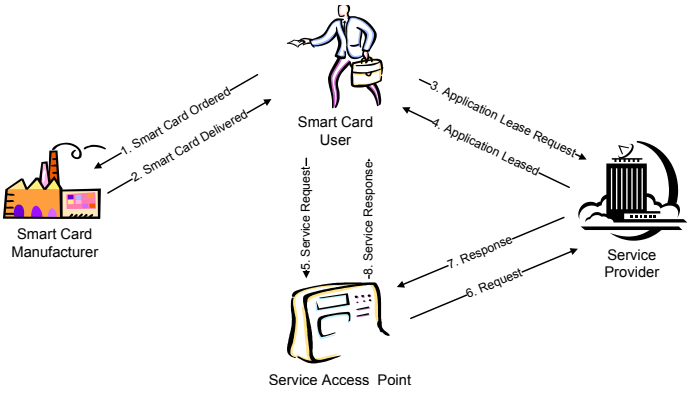

Figure 2. User Centric Smart Card Ownership Model

Although a UCOM enabled smart card could be in any form factor, in course of this paper we would consider the configuration in which it is a secure element in an NFC enabled smart phone. This configuration provides the highest level of dynamic and ubiquitous access to several services than the other configurations like a standalone smart card (i.e. credit card form factor). In the later configuration, a user might have to use additional hardware like a smart card reader, computer or an access point (like an ATM) to install or delete an application on her smart card.

B. Smart Card Architecture

The proposed architecture for a UCOM smart card is depicted in figure 3 and this architecture satisfies the requirements of the UCOM discussed in [5].

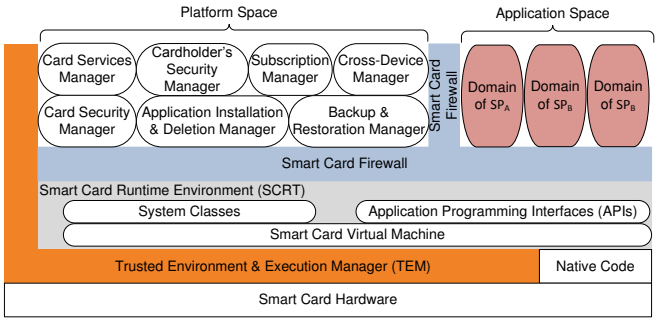

Figure 3. User Centric Smart Card (UCSC) architecture

Most of the components shown in figure 3 are either an improvement to the existing framework or an addition to the GlobalPlatform architecture. We use GlobalPlatform as the base architecture for the components in this section. These components modify the GlobalPlatform card specification to accommodate the UCOM philosophy. For brevity, we will only discuss the backup & restoration manager and Trusted Environment & Execution Manager (TEM) in this paper.

1) Backup & Restoration Manager: The interface to the proposed mechanisms in this paper is referred as Backup & Restoration Manager (BRM). The BRM is implemented by the platform designers (card manufacturers) and it communicates with the external entities like backup server or other smart cards. No sensitive data is stored in BRM, which is actually stored securely in the TEM discussed in the next section.

2) Trusted Environment & Execution Manager: On a typical smart card, several mechanisms are in place to test and verify the state of the platform (both software and hardware). At the software level, GlobalPlatform card specification has proposed the controlling authority (termed CA in the GlobalPlatform card specification) [11] and the Mandated Data Authentication Pattern (Mandated DAP) mechanism [11], [12]. In the DAP mechanism, an off-card entity (controlling authority) signs applications that are being loaded onto a smart card, and this approval of the applications is verified by an on-card entity referred to as the GlobalPlatform card manager [12]. At the hardware level, the Known Answer Test (KAT) for cryptographic modules mandated by FIPS [13] and similar mechanism are deployed by the smart card manufacturer (i.e. RAM test, and checking checksum of non-volatile memory, etc.) [14].

The Trusted Computing Group (TCG) has initiated a working group to devise specifications for a trusted module for embedded devices [15]. We propose the Trusted Environment & Execution Manager (TEM) as a trusted module for embedded devices like smart cards. The TEM is fundamentally different from the Trusted Platform Module (TPM) [16] and Mobile Trusted Module (MTM) [17] in two respects. Firstly, the TEM implements a self-test mechanism that includes hardware parameters to provide remote attestation and a dynamically configurable integrity measurement mechanism that is based on a challenge-response framework. Secondly, the TEM is not based on a static architecture; in fact, it enforces platform security policies during the application execution rather than just generating the hash (once) at the start of the application execution. The architecture of the TEM is illustrated in figure 4.

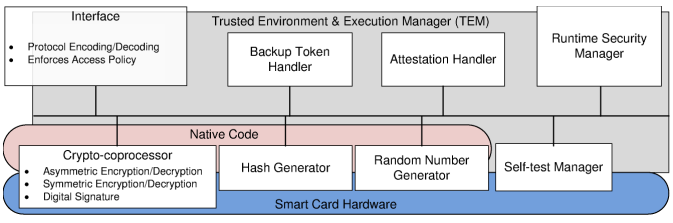

Figure 4. Architecture for the Trusted Environment & Execution Manager

The concept of TEM is to group/provide similar and enhanced functionality that provides assurance and validation of the platform to requesting on-card or off-card entities. The TEM is independent of the platform configuration that is mainly concerned with the smart card runtime environment, which can be based on a technology such as Java Card [18] or Multos [19]. A TEM does not have to be implemented in hardware; it can be software-based and utilise the smart card's cryptographic hardware (the crypto co-processor). The TEM requires access to the crypto co-processor for encryption/decryption, signature generation and verification, and random number generation. In this paper, we will not detail the design of the TEM and restrict the discussion only to the Backup Token Handler discussed in next section.

a) Backup Token Handler: The backup token handler acts as a secure repository that stores the restoration tokens of individual applications (if sanctioned by their respective SPs) and sensitive data associated with the BRM. When a user registers with a Secure Backup Server (SBS) or wants to transfer the installed applications from one smart card to another, the BRM retrieves these tokens from the backup token handler, encrypts them, and communicates to the intended entity (e.g. SBSor new smart card). The details of this mechanism are further elaborated in section III.

C. Motivation

The UCOM at one end facilitates a user to have most, if not all of her applications onto a single smart card. This could help the user to perform mundane tasks of modern life, without a great deal of hassle that one has to put up when dealing with a large number of different smart cards. Equally it also creates the issue that by having all of the identities (applications) on one chip increases the adverse effect if it is being stolen or lost. The traditional procedure to get a replacement card in case of theft or loss takes days or even weeks in some cases. To align the recovery process in the UCOM, we propose a secure, reliable, ubiquitous and on-demand recovery framework paper.

The UCOM enables a user to have most services that she is entitled to use, including banking, transport, access control, health and loyalty on her smart card. In such a scenario, most of the essential services that are necessary to perform even the mundane tasks might be on a single chip. Therefore, in adverse circumstances of losing her smart card, a backup mechanism will enable her to retrieve and restore the contents on to a new smart card. In addition, this mechanism can also (implicitly) block the applications on the stolen card from accessing the associated services from their respective SPs. To have a backup, ideally it is recommend to be stored some place safe (i.e. a trusted third party) from where in case of emergency it could be retrieved. Traditionally, in the high-end computing environments (e.g. servers and personal computers) entire applications along with operating systems can be backed up. However, such a mechanism may not be suitable for the smart card devices for the reasons listed below.

1) Applications on a smart card are considered as secure access tokens to associated services provided by their respective SPs. The smart card acts as an additional security mechanism to authenticate the respective user to the SP.
2) An application downloaded to a smart card is bounded to it [9] and changing that without notifying the respective SP would violate the terms and conditions of the ALP.

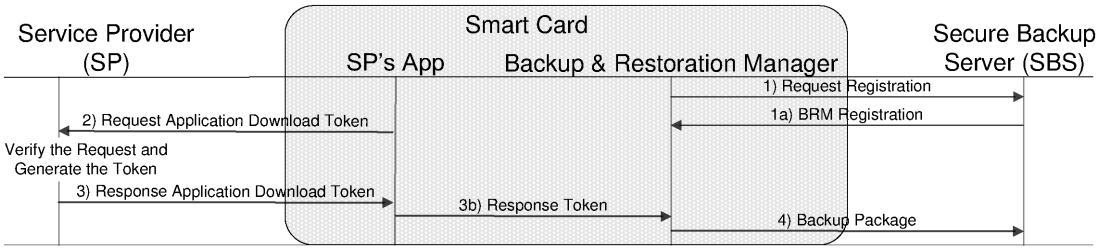

Figure 5. Overview of Backup Mechanism

3) There is a possibility that the remote backup location might not be tamper-resistant and a malicious user could take advantage of it.
4) Backing up entire applications might lead to the possibility of application cloning by a malicious users.
5) A malicious user could simulate the smart card environment copy the whole application that is backed up in order to reverse engineer it — retrieving sensitive data and algorithms implemented as part of the application. This issues is discussed in depth in [20].

Therefore, it would be safe to assume that instead of taking a backup of the applications, it would be rational to backup the application download credentials (i.e. authorisation token). These are the credentials which would be issued by the SPs to their respective users, so in case of stolen or lost card the user could use these credentials to initiate an automatic application download process. Such a mechanism is the core aim of this paper.

III. Backup and Migration Framework

In this section, we describe two mechanisms: backup and migration. In the backup process, a user archives her smart card's contents (i.e. authorisation tokens) to a backup server, which can be used to restore her contents to the destination smart card - if such a need arises. In the migration process, there is no backup server and the smart card contents are transferred between a source and a destination smart card.

A. Backup Mechanism

In the backup mechanism the authorisation tokens issued by SPs to their respective users are stored as a "backup package" at a secure location, preferably accessible ubiquitously on-demand. When a user wants to restore the contents of her old smart card, she has to import the backup package; then the individual applications will be requested from their respective SPs automatically by the BRM of the new smart card using the associated authorisation tokens.

In our proposal, a secure off-site backup facility is provided by a secure third party referred to as a Secure Backup Server (SBS). We do not consider that a SBS has to be an SP and the only requirement is that users trust the SBS. A backup framework overview is illustrated in figure 5 and described below.

1) A smart card user registers herself to a SBS using the Secure and Trusted Channel Protocols (STCPs) proposed in [21]. After the registration, the BRM has the user's credentials and details of how to connect with the respective SBS. The BRM and SBS will mutually generate a shared secret that they will be used in future sessions. As this shared secret is bound to the specific smart card, it is only used for secure communication and not sealing (encrypting) the backup package.
2) After an application is installed on a smart card, it can initiate the request for an authorisation token. The application will only request for the authorisation token if it is sanctioned by the respective SP. We opted for two possible scenarios: restorable and non-restorable applications. These types are inspired by the security policy related to key migration in the TPM specification [16]. For restorable applications, an SP will issue its application with an authorisation token, and the (host) smart card would only migrate this token to the destination smart card or a SBS: for non-restorable, the respective SP will not issue any authorisation token.
3) An SP sends its installed application the authorisation token (if it opts for it) that consists of two sections as shown in figure 6. The first section is a public section that is not encrypted and it contains the SP's URL (Universal Resource Locator), authorisation token identifier, and on optional section. The URL would instruct a smart card where to establish the connection to download the application. The authorisation token identifier uniquely identifies the token and associated cryptographic keys. The optional segment is made available for the SP to include any housekeeping information if necessary. The second section consists of an encrypted message that may contain proprietary information that would ensure that the token is genuine and is generated by the SP. Tis section is encrypted by the SP with its token authorisation key and the selection of this key is at the sole discretion of the SP. The contents of this section include an application identifier, a user identifier and an application lease identifier. The application identifier refers to the application that was issued to the user indicated by the user identifier. The application lease identifier uniquely identifies the previous smart card to which the application was leased, along with any associated data, including cryptographic keys (if each instance of the application lease has different cryptographic keys).

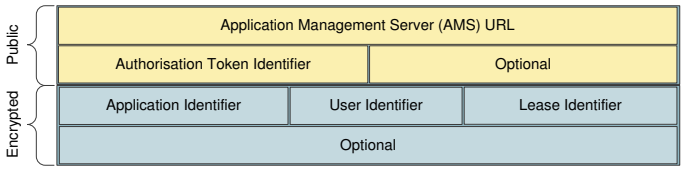

Figure 6. Structure of authorisation tokens generated by respective SPs

Table I
PERFORMANCE MEASURES (MILLISECONDS).

Measures	Card Two	Card One	Memory Usage
Token Acquisition	3198.71	3291.38	9856
Uploading Package	3213.57	3301.21	9892
Migration	3056.69	3193.58	8563
Application Restoration	3396.65	3526.32	9918

Note: Memory usage is associated with storage space used on the test Java Cards and it is measured in bytes..

4) On receipt of the authorisation token, the respective application will forward it to the BRM. The BRM will encrypt the set of authorisation tokens with a package sealing key that is based on some secret that is known to the user. It could be a password, a passphrase or a biometric — something that the user could provide at the time of restoration to prove that she is the genuine user that created the backup package. This key would be generated once, unless the user decided to change her password or passphrase. The simplest way to generate the package-sealing key is to base it on the user's input. The size of the key and password length is based on the user's choice.

At the time of restoration, the user will provide the BRM of the new smart card with the credentials for the SBS. The BRM and SBS will establish a secure relationship using the STCP. Subsequently, the BRM will download the backup package (authorisation tokens) from the SBS. These authorisation tokens are sealed by an encryption key based on the user's input. The BRM will request the user for the relevant input and decrypt the backup package. After decryption of the package, the BRM will retrieve one authorisation token at a time and use its public section to connect with the associated SP. To establish a secure channel and authenticate the user to the given SP, we modify the STCP$_{SP}$ (discussed in [21]). In the fourth message of the STCP$_{SP}$, we replace the U$_{Cre}$ with the authorisation token issued by the SP.

Before an SP issues a new lease to the user, it terminates the existing lease. This means that although the lost smart card is still usable, the malicious user cannot utilise the application on it to access sanctioned services. There are two protection mechanisms that avoids the unauthorised use of the applications on the lost or stolen card. The first mechanism is on-card protection based on the Personal Identification Number (PIN) verification (if implemented). If an application requires PIN verification before it executes, the usual protection mechanism that disables a smart card (or application) if the user enters the wrong PIN multiple times will suffice. Whereas, in the second mechanism, the SP can simply blacklist the application from access the associated services. If the application tries to access the services, the SP can instruct the application to block itself and if possible delete all data related to the particular lease and user — only if the lost/stolen smart card tries to access the SP's services. One point to note is that in the UCOM, an SP can only block and delete its application, and cannot block/lock the smart card. Nevertheless, an adversary can still use an application only if it does not require a PIN verification and live connection with the respective SP when it executes.

B. Migration Mechanism

In the previous section, we discussed the structure of an authorisation token and framework for backup to a remote server (e.g. SBS). In this section, we use the same authorisation tokens but this time for migrating contents from one smart card to another.

Similar to the key migration in the TPM specification [16], when a user initiates an application migration process. The TEM of the source smart card establishes a secure channel with the destination smart card via BRMs, using the Platform Binding Protocol discussed in [22]. The destination smart card then requests the transfer of the authorisation tokens from the source smart card. The migration process first deletes applications from the source smart card, then transfers the authorisation token to the destination smart card. The applications will be downloaded on to the destination smart card in a manner similar to the one discussed in the previous section. This process is similar to the TPM key migration, except we use a different protocol to the one specified by the TPM specification [16].

IV. IMPLEMENTATION AND PERFORMANCE MEASUREMENT

For performance measurements, we use the test bed consisting of a laptop with 1.83 GHz, and 2GB RAM running on Windows XP along with two sets of 16bit Java Cards. For the emulation of the backup mechanism, the BRM, TEM and smart card application are implemented on the Java Card where the laptop takes the role of the secure SBS and SP. Where in the case of the migration mechanism, the laptop just acts like a communication bridge between the two Java Cards and each card in a set takes the role of the source and destination cards. The performance measures listed in the table I includes acquiring restoration tokens, uploading backup package to the secure SBS, migration between two cards and application restoration. In the test bed implementations, for the cryptographic algorithms, we have selected Advance Encryption Standard (AES) [23] 128-bit key symmetric encryption with Cipher Block Chaining (CBC) [24] without padding for both encryption and MAC operations. The signature algorithm is based on the Rivest-Shamir-Aldeman (RSA) [24] 512-bit key. We used SHA-256 [25] for hash generation. For Diffie-Hellman key generation we used a 2058-bit group with a 256-bit prime order subgroup specified in the RFC-5114 [26].

For the acquisition of the restoration token and key generation & uploading of the backup package, we deployed a slightly modified STCP discussed in [27]. For the migration mechanism, we used the Platform Binding Protocol discussed in [22]. For application restoration, as discussed in section III-A we modified the protocol proposed in [21]. All the protocols that we deployed for each stage of the implementation were slightly modified (in their implementation) to include the restoration token and backup package.

V. Analysis of the Backup and Migration Mechanism

In the smart card industry, there are not many examples of contents backup or migration mechanisms that we can compare with ours. An example is the backup mechanism for phone-book contacts, but even this mechanism is not like the one discussed in this paper. Although not the same, but the TPM key migration architecture [16] can be regarded closest our proposal. The application migration process is similar to the TPM key migration and the only difference is that instead of migrating keys, we migrate the authorisation tokens to the destination smart cards. In the smart card industry such mechanisms are not required due to the ICOM architecture.

The contents backup mechanism effectively prevents smart card cloning and intellectual property theft. In smart card cloning, a malicious user tries to copy applications from a smart card to another card, without the permission of the respective SPs. To prevent cloning of an application, the relevant SP is given the ability to make its application either restorable or non-restorable. Therefore, the choice of moving the application to a new smart card is not with the user but with the SP. Furthermore, the backup or migration mechanism does not move the application data or/and code. In fact, even when the SP sanctions its application to be restorable, the mechanism still relies on the SP to issue an authorisation token. Without this authorisation token, the application cannot be part of the backup or the migration mechanism.

Intellectual property theft refers to the scenario where a malicious user tries to obtain the application code (along with data). To do so, the malicious user has to access the application on a non tamper-resistant device with minimal protection. Such a scenario can arise if we move the entire application (code and data) off-card during the backup or migration mechanisms. Therefore, by using authorisation tokens the backup and migration mechanism effectively prevent intellectual property theft.

In addition, the lease of the application to the destination smart card is at the sole discretion of the SP. Therefore, after evaluating the operational and security capabilities of the destination smart card, the SP can continue and lease its application. Furthermore, the SP could first block the lease of the previous application before leasing to the new smart card. Nevertheless, there are certain concerns in the contents backup mechanism that are related to the key that encrypts/decrypts the backup packages. The framework requires the user to input a secret value that could be a long PIN, password, or passphrase that can be exploited by an adversary. To avoid the use of weak user passwords it is recommended the SBSs should take adequate measures by requiring users to choose strong passwords. Furthermore, before a user can download authorisation tokens from the SBS there should be some offline authorisation (e.g. activation of restoration process on a SBS over the internet or telephone).

The migration mechanism is similar to the backup mechanism, except for one detail. It does not require a SBS, so it avoids the need for user password-based cryptographic keys. We consider that the BRM of a given smart card should support both the backup and migration mechanisms.

VI. Conclusion

In this paper, we briefly described the User Centric Smart Card Ownership Model (UCOM) and contrasted it with the traditional Issuer Centric Smart Card Ownership Model. The open and dynamic nature of the UCOM enables a user to have multiple applications on her smart cards which both increase the adverse due to either damage, lost or theft of the smart cards.

We proposed a smart card contents backup and migration mechanism, which enables a card user to backup/migrate her applications if required. The backup and migration mechanism does not move the applications out of the secure smart card storage location — in fact they only retrieve the application credentials that a user can utilise in future to download the application to her new smart card. Subsequently, we analysed the implementation of the proposed backup and migration mechanism.

In the smart card technology, such a mechanism is not defined in its current state. Similar mechanisms can be argued to exist in the (U)SIM environment but they only backup the phone-book — which is not similar to the backing up the applications from a smart card. Therefore, the proposal presented in this paper is a unique of its kind in the smart card industry.

References

[1] K. Mayes and K. Markantonakis, Eds., *Smart Cards, Tokens, Security and Applications*. Springer, 2008.

[2] *ISO/IEC 18092: Near Field Communication - Interface and Protocol (NFCIP-1)*, International Organization for Standardization (ISO) Std., April 2004.

[3] "Mobile NFC Services," GSM Association, White Paper Version 1.0, 2007. [Online]. Available: http://www.gsmworld.com/documents/nfc_services_0207.pdf

[4] "GlobalPlatform's Proposition for NFC Mobile: Secure Element Managment and Messaging," Online, GlobalPlatform, Specification, April 2009.

[5] R. N. Akram, K. Markantonakis, and K. Mayes, "A Paradigm Shift in Smart Card Ownership Model," in *Proceedings of the 2010 International Conference on Computational Science and Its Applications (ICCSA 2010)*, B. O. Apduhan, O. Gervasi, A. Iglesias, D. Taniar, and M. Gavrilova, Eds. Fukuoka, Japan: IEEE Computer Society, March 2010, pp. 191–200.

[6] "Pay-Buy-Mobile: Business Opportunity Analysis," GSM Association, White Paper 1.0, November 2007. [Online]. Available: http://www.gsmworld.com/documents/gsma_nfc_tech_guide_vs1.pdf

[7] S. Chaumette and D. Sauveron, "New Security Problems Raised by Open Multiapplication Smart Cards." *LaBRI, Université Bordeaux 1.*, pp. 1332–04, 2004.

[8] "A New Model: The Consumer-Centric Model and How It Applies to the Mobile Ecosystem." GlobalPlatform, March 2012, Whitepaper.

[9] R. N. Akram, K. Markantonakis, and K. Mayes, "Application Management Framework in User Centric Smart Card Ownership Model," in *The 10th International Workshop on Information Security Applications (WISA09)*, H. Y. YOUM and M. Yung, Eds., vol. 5932/2009. Busan, Korea: Springer, August 2009, pp. 20–35. [Online]. Available: http://www.springerlink.com/content/f7027021h1067261/fulltext.pdf

[10] D. Sauveron, "Multiapplication Smart Card: Towards an Open Smart Card?" *Inf. Secur. Tech. Rep.*, vol. 14, no. 2, pp. 70–78, 2009.

[11] "The GlobalPlatform Proposition for NFC Mobile: Secure Element Management and Messaging," GlobalPlatform, White Paper, April 2009. [Online]. Available: http://www.globalplatform.org/documents/GlobalPlatform_NFC_Mobile_White_Paper.pdf

[12] *GlobalPlatform: GlobalPlatform Card Specification, Version 2.2,*, GlobalPlatform Std., March 2006. [Online]. Available: http://www.globalplatform.org/specificationscard.asp

[13] *FIPS 140-2: Security Requirements for Cryptographic Modules*, Online, National Institute of Standards and Technology (NIST) Federal Information Processing Standards Publication, Rev. Supercedes FIPS PUB 140-1, May 2005. [Online]. Available: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[14] W. Rankl and W. Effing, *Smart Card Handbook*, 3rd ed. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[15] (Visited September, 2011) Trusted Computing Group: Embedded Systems Work Group. Online. Trusted Compouting Group. Oregon, USA. [Online]. Available: http://www.trustedcomputinggroup.org/developers/embedded_systems

[16] *Trusted Module Specification 1.2: Part 1- Design Principles, Part 2- Structures of the TPM, Part 3- Commands*, Trusted Computing Group Std., Rev. 103, July 2007. [Online]. Available: http://www.trustedcomputinggroup.org/resources/tpm_specification_version_12_revision_103_part_1__3

[17] "TCG Mobile Trusted Module Specification," Trusted Computing Group (TCG), Specification Ver 1.0, June 2008.

[18] *Java Card Platform Specification: Classic Edition; Application Programming Interface, Runtime Environment Specification, Virtual Machine Specification, Connected Edition; Runtime Environment Specification, Java Servlet Specification, Application Programming Interface, Virtual Machine Specification, Sample Structure of Application Modules*, Oracle Std. Version 3.0.1, May 2009. [Online]. Available: http://java.sun.com/javacard/3.0.1/specs.jsp

[19] *Multos: The Multos Specification,*, Online, Std. [Online]. Available: http://www.multos.com/

[20] R. N. Akram, K. Markantonakis, and K. Mayes, "Simulator Problem in User Centric Smart Card Ownership Model," in *6th IEEE/IFIP International Symposium on Trusted Computing and Communications (TrustCom-10)*, H. Y. Tang and X. Fu, Eds. HongKong, China: IEEE Computer Society, December 2010.

[21] ——, "A Secure and Trusted Channel Protocol for the User Centric Smart Card Ownership Model," in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-13)*. Melbourne, Australia: IEEE Computer Society, 2013.

[22] ——, "Application-Binding Protocol in the User Centric Smart Card Ownership Model," in *(to appear in) the 16th Australasian Conference on Information Security and Privacy (ACISP)*, ser. LNCS, U. Parampalli and P. Hawkes, Eds. Melbourne, Australia: Springer, July 2011, pp. 208–225.

[23] Joan Daemen and Vincent Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Berlin, Heidelberg, New York: Springer Verlag, 2002.

[24] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC, October 1996.

[25] *FIPS 180-2: Secure Hash Standard (SHS)*, National Institute of Standards and Technology (NIST) Std., 2002. [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[26] M. Lepinski and S. Kent, "RFC 5114 - Additional Diffie-Hellman Groups for Use with IETF Standards," Tech. Rep., January 2008. [Online]. Available: http://tools.ietf.org/html/rfc5114

[27] R. N. Akram, K. Markantonakis, and K. Mayes, "Building the Bridges – A Proposal for Merging different Paradigms in Mobile NFC Ecosystem," in *The 8th International Conference on Computational Intelligence and Security (CIS 2012)*, S. Xie, Ed. Guangzhou, China: IEEE Computer Society, November 2012.