Rethinking the Smart Card Technology

Raja Naeem Akram[1] and Konstantinos Markantonakis[2]

[1] Cyber Security Lab, Department of Computer Science, University of Waikato, Hamilton, New Zealand.
[2] ISG Smart card Centre, Royal Holloway, University of London. United Kingdom
rnakram@waikato.ac.nz, k.markantonakis@rhul.ac.uk

Abstract. Creating security architectures and processes that directly interact with consumers, especially in consumer electronics, has to take into account usability, user-experience and skill level. Smart cards provide secure services, even in malicious environments, to end-users with a fairly straightforward limited usage pattern that even an ordinary user can easily deal with. The way the smart card industry achieves this is by limiting users' interactions and privileges on the smart cards they carry around and use to access different services. This centralised control has been the key to providing secure and reliable services through smart cards, while keeping the smart cards fairly useable for end-users. However, as smart cards have permeated into every aspect of modern life, users have ended up carrying multiple cards to perform mundane tasks, making smart card-based services a cumbersome experience. User Centric Smart Cards (UCSC) enable users to have all the services they might be accessing using traditional smart cards on a single device that is under their control. Giving "freedom of choice" to users increases their privileges, but the design requirement is to maintain the same level of security and reliability as traditional architectures while giving better user experience. In this paper, we will discuss the challenges faced by the UCSC proposal in balancing security with usability and "freedom of choice", and how it has resolved them.

1 Introduction

A smart card is a small, resource-restricted and highly security-sensitive device whose fundamental goal is to enable secure services for its users. These devices have been deployed in a large number of heterogeneous industries and used by a huge user base. A smart card has an embedded device which is part of the plastic body of credit cards and SIM cards. The inception of smart cards is rooted in the need to create a highly secure device that is then issued to users, some of whom could be malicious while others may be technologically naive. These represent the two extremes of user competence/knowledge of smart card technology. Since the 1970s, the smart card industry has created successful devices that satisfy the core requirement: a product that is intuitively simple but at the same time has high security assurance[3] - even in the possession of malicious users.

[3] Smart cards in certain industries like banking have stringent security requirements, including a detailed third party evaluation based on Common Criteria (CC) [1,2]. In

To balance the security requirements of a particular application and its usability is difficult at best [4]. An application (or device) in the possession of a malicious user makes balance difficult to achieve [5]. The assumption that an increase in usability might negatively affect the overall security of digital systems is not an exaggeration. Along with maintaining security and tamper-resistance while in the possession of a malicious user, a smart card also has to be designed in a manner whereby normal users don't have to perform complicated tasks [6]. An example is the number of steps a user might have to take to access an encrypted/signed email service. Johnny of Whitten and Tygar [7] was troubled by the complicated and technology-intense tasks that he had to perform to achieve the required security goal (i.e. encryption). In smart card deployments, users are not required to perform complicated tasks except for banking [8] or access control [9] applications. In banking and access control applications a user might be required to enter a four (or more) digit Personal Identification Number (PIN). Aside from this input, the user does not have to do anything extra: the smart card then performs the security-related tasks in a seamless manner [10].

To provide a high level of security and require the least user interactions to achieve this, the smart card industry preferred the Issuer Centric Smart Card Ownership Model (ICOM) [11]. The ICOM model enables a centralised authority to manage and issue smart cards to users. Examples of centralised authorities include telecom, banking and transport companies, also referred to as card issuers. Card issuers provide services to their customers via their smart cards; therefore, smart cards act like a secure token that give them access to available services. These card issuers maintain and manage the security features of the smart cards and in most cases do not require the user to perform any technologically challenging tasks (e.g. SIM cards in most of mobile phones) [12].

However, since 2005 technologies like smartphone "Apps" [13] and Near Field Communication (NFC) [14] have changed the smart card technology landscape. Furthermore, Johnny of today requires more features present on a single device. Smart cards can support multiple applications [15] on a single device, but such an initiative did not initially achieve widespread deployment. However, with the advent of NFC and the Apps culture, different organisations have proposed a multiple application smart card initiative termed the Trusted Service Manager (TSM) [16,17]. In addition to the TSM, there are other initiatives including our proposal, the User Centric Smart Card Ownership Model (UCOM) [18]. Furthermore, a model similar to the UCOM has been proposed by GlobalPlatform termed the "Consumer-Centric Model" [19]. In this paper, we discuss the usability and security considerations that we took into account when designing the UCOM.

1.1 Structure of the Paper

In section 2, we discuss the *open card* initiative which was one of the first attempts to offer users "freedom of choice". In subsequent sections, we briefly de-

contrast, while smart cards play a crucial role in security for mobile telecom, they do not require CC evaluation [3].

scribe the UCOM and user requirements that became the core of the UCOM design. Section 4 details selected operations of the UCOM to show how the principle of least interaction is used in practice. Finally, in section 5 we conclude the paper.

2 Open Cards

In this section, we briefly discuss the open card initiative and concerns about the usability of this proposal.

2.1 Brief Introduction

It is difficult to give an exact definition of open cards. In general, however, the term "open card" is used to refer to blank smart cards that a user can purchase from a supplier. After purchasing the smart card, the user can perform the role previously performed by the card issuer and either accept or buy applications from different application providers. These applications can be installed onto the user's card and used to access any associated services. The whole card is under the user's control, similar to the card issuer in the ICOM. Therefore, we can say that the open card initiative is an ICOM framework with the user replacing the card issuer.

Traditional smart card frameworks like Java Card, Multos, and GlobalPlatform were considered suitable for such a scenario. Most of these frameworks were built to support the ICOM, and by making the user an issuer, they did not require any substantial changes. However, as implied by Pierre Girard [20], such a mechanism would require an application provider to issue their application to users to install on their smart card. This would require the application provider to trust the user not to reverse engineer or corrupt the application.

Such a scenario does not ensure the security, protection of intellectual property, and reliability of an application, as an application provider does not have any control over the smart card that hosts its application. The main reason for this lack of control on the part of the application provider is the unavailability of any guarantees regarding the security and operational behaviour of smart cards. Similar security issues are raised by Chaumette and Sauveron in [21] and they make the open card initiative in its current form unsuitable for a user-centric framework.

2.2 Issues with Open Card Model

In this section, we will only discuss issues related to the open card model from the usability and least interaction point of view. As discussed in the previous section, the open card model gives a user the ability to download an application to their device of choice (e.g. desktop or laptop). Once the application is downloaded to the user's device, she can then transfer the application to her smart cards. The issue is transferring the application to the smart card: anyone who has worked

with installing applications on embedded devices knows that such a task is not trivial. Furthermore, from a security point of view the user has to ensure that during this process no malicious entity can corrupt the application. The user has to perform several tasks and ensure the safe transfer of the application to the smart card, increasing rather than decreasing user interaction. In the UCOM, the least interaction principle requires the user to either not be involved or if required, her involvement to be restricted to the minimum level possible.

3 User Centric Smart Card Model

In this section, we briefly discuss the core design of the UCOM and associated user requirements that became the basis of our subsequent rethinking of smart card technology.

3.1 User: The Core of Design

A user acquires a User Centric Smart Card (UCSC) from a UCSC supplier, and then manages it through software referred to as Card Application Management Software (CAMS): shown in figure 1. The CAMS only provide an interface with the UCSC and there are no security requirements for it (i.e. as part of the design we consider that the CAMS implementation can be modified by a malicious user).

The user can then request a Service Provider (SP): an application provider that utilises the UCSC functionality to provide a secure, reliable and privacy-preserving service. The SP will then request the security and reliability verification and validation of the UCSC [22]. Only after the SP is satisfied with the security and functional-support of the UCSC it will lease its applications. The application lease is governed by a security and functional-support policy of the SP, referred to as an Application Lease Policy (ALP) [11]. The ALP is an SP-specific document and an SP can reject a request for application lease if the requesting UCSC does not support the SP's ALP. Once the application is leased to the UCSC, it can be accessed by the user at any compatible computing platform shown as a Service Access Point (SAP)/Host Platforms in Figure 1.

For the smart card environment, a downloaded application might be a stand-alone application that does not require any accompanying application on the host platform. In the case of a smart card environment, the host platform is the card reader that communicates with the smart card. The reader needs to have an application (of its own) that communicates with the smart card but this requirement is not imposed by the smart card's applications, and is installed separately by the entity that maintains the reader. For example, in the banking and telecom sector the reader only has to conform to a standardised application (e.g. EMV [8]); however, in the transport-service scenario it varies, as different operators install their own readers with customised applications (i.e. TFL [23] and Octopus [24]). However, in case of hand-held and traditional computing devices, applications installed on a UCSC might be part of a larger application that is actually installed on the host platform.

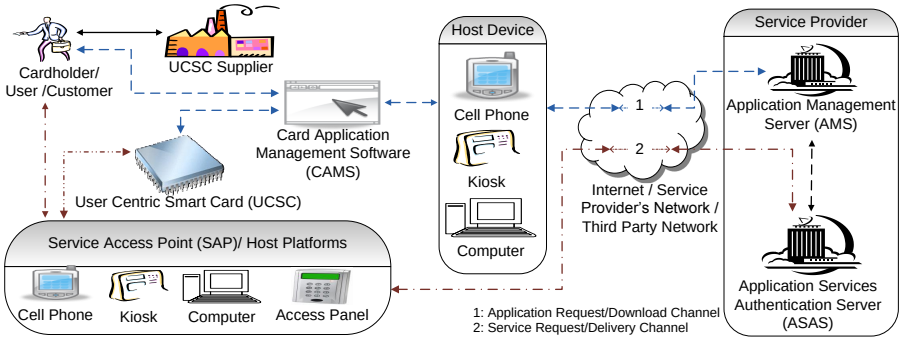

Fig. 1. User Centric Smart Card Framework

3.2 User Requirements

A cardholder is an entity that uses a smart card to access authorised services. In the UCOM, the control of a smart card is with its user. Therefore, cardholders have complete control over the choice of applications on their smart cards. They have the flexibility to change the installed applications on their smart cards. Furthermore, they can install or delete any applications they are entitled to, at their convenience. The framework will provide the mechanism that ensures secure control and ubiquitous management of applications on smart cards. A cardholder's requirements in UCOM are listed below:

1. **Security:** If a smart card is inherently insecure, or if it becomes vulnerable to new threats, it can affect the security of applications installed on the card. We cannot expect that each cardholder is technically capable of ensuring and managing the security of the smart card; therefore, a cardholder would require an assurance that the card platform will be secure and reliable even if it is in the possession of a technologically naive or malicious user.
2. **Privacy:** Applications installed on a smart card represent the identities of the cardholder in different contexts. For example a college card, a health card and a credit card represent a cardholder's identity as a student, a patient, and a consumer respectively. These identities are in the form of applications that have some unique characteristics (e.g. student ID, patient ID, and Primary Account Number: PAN) to identify a particular user. Therefore, applications on a smart card can be treated as the identities of the cardholder. In the ICOM, these identities may not have any connection with each other. However, in the UCOM, any or all of these identities could be on the same card, creating a privacy issue if one application becomes aware of the existence of others on a smart card. Therefore, the identities on a particular card should not have any links between them. For example, a college application should not be able to find out about a medical application(s) installed on the same card.

3. **Least Interaction (Seamless Framework):** Most users do not understand the technology behind a particular product (i.e. mobile phone applications). Therefore, the framework should not be based on the assumption that an average user can perform technically challenging tasks. The UCOM should be seamless and should perform all necessary tasks by itself, only involving the user when required.
4. **Interoperability:** The smart card user will not want to buy a separate smart card for each application. Smart card suppliers should provide cards that support most of the available functionalities and SPs should offer applications in many formats as possible, to support a range of different execution environments.
5. **Ownership Mechanism:** A mechanism is required that securely authenticates the owner of the smart card and facilitates the exercise of her privileges (i.e. installing and deleting applications).

4 Designing Security for Malicious and Tech-Illiterate Users

In this section, we explore a few of the UCOM operations to show how a secure system can be designed based on minimal user interaction.

4.1 Usability and Security

Selected UCOM operations that had to take into account the security and usability are: User Ownership Acquisition, Application Installation, Application Sharing and Decommissioning Process. Crucially these operations are managed by the card issuer in the ICOM without any user input. However, by giving "freedom of choice" to the user in the UCOM, the outcome of these operations affect the user's device.

4.1.1 User Ownership Acquisition A UCSC in its pre-issuance state is under the default ownership of the UCSC manufacturer. When a user takes control of the smart card, it will initiate an ownership acquisition process. The process is described below:

1. The user initiates the ownership acquisition process through the Card Application Management Software (CAMS) shown in Figure 1.
2. The UCSC requests the default ownership credentials, which are communicated to the user by the card manufacturer. In response, the user will provide the relevant default credentials.
3. On verification of the credentials, the UCSC checks the mode of platform assurance and validation selected by the user. The supported modes are offline and online attestation [25,26]. Depending upon the user's choice the UCSC proceeds with the security attestation process.

4. Once the assurance validation is communicated to the CAMS, the user can compare the smart card features with those stated by the card manufacturer at the time of purchase. If satisfied, the user will provide her credentials and they are used to authenticate the user to the UCSC for management operations (e.g. application installation, and deletion). The credentials can be based on a Personal Identification Number (PIN), a password, a passphrase, or biometric data [27] depending upon the card manufacturer, and the user's requirements.

The decommissioning process (section 4.1.4) is used when a user relinquishes control of a UCSC to re-sell or scrap the device. The process is similar to ownership acquisition but this time the user requests ownership delegation that will delete the user's space and any applications she has installed in it.

4.1.2 Application Installation In this section, the processes that support the secure transmission and installation of an application are discussed. The installation process discussed in this section builds additional checks around the application installation protocols [28,29,30].

The installation request will initiate the process of acquiring an application from an SP's application server (AMS in figure 1) and installing it on a smart card. The entire process can be divided into three sub-processes: 1) Downloading, 2) Localisation, and 3) Application Registration. These sub-processes are explained as below.

1. Downloading: The downloading of an application is initiated by the smart card, through a secure channel protocol [28,29]. At the conclusion of the secure channel protocol, both entities generate a set of keys for application download and domain management. The smart card then generates an SP's domain, provided it has enough space to accommodate it. The SP and smart card will then start the application downloading process. The SP will first generate a signature on the application, then encrypt and MAC it before sending it to the smart card.
   The smart card checks the generated MAC, decrypts the application, and verifies the signature. A decrypted application is not a fully installed application — it is the equivalent of copying an application to a memory location. The next step is to verify whether the application complies with the smart card's operational and security policy. For this purpose an on-card byte code verification is performed [31], which is already mandated by the Java Card 3 [32]; this can be based on well-defined on-card byte code verification proposals [33].
   The UCSC does not mandate the security evaluation of an application. However, certain applications require evaluation due to government or industry regulations (e.g. EMV applications). In these cases, an SP's application(s) provide an evaluation certificate [22]. To verify the certificate the smart card would have to calculate the hash of the downloaded application and compare it with the Application Assurance Certificate (AAC) [22].

2. Localisation: First, the application will be personalised by the SP. Depending upon the relationship between the cardholder and the SP, with the SP's discretion the personalisation can include acquiring user details (in post- and no-registration scenarios), and cryptographic key generation. Furthermore, if the SP is issuing a card-bound lease then it would make sense to generate on-card cryptographic keys. These keys will automatically become device identifiers because each lease of the application will have a specific set of keys. After personalisation, the downloaded application establishes connections with various on-card services (i.e. shareable resources) that are provided by partner applications. To access a partner's application services, the downloaded application will establish an application-sharing relationship that is discussed in detail in [34,35].
3. Application Registration: The final stage of an application installation is application registration by the SP. Registration allows the application to access sanctioned services. Once the SP registers (sanctions) the downloaded application, the smart card will also make it selectable to an off-card entity. By making an application selectable, the smart card allows the application to execute and access on-card services and communicate with off-card entities.

4.1.3 Application Sharing In this section, we discuss the architecture of the proposed firewall mechanism for UCSCs. The proposed firewall mechanism is based on the Java Card firewall mechanism as illustrated in Figure 2 that is discussed subsequently.

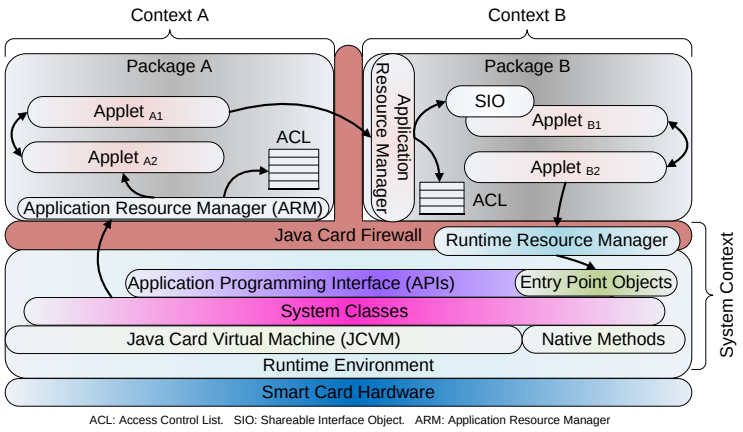

Fig. 2. Architecture of the UCTD firewall mechanism

A request for an application's shareable resource is handled by the application's Application Resource Manager (ARM) and the Runtime Resource Manager (RRM) handles access to the platform's resources (APIs): see figure 2.

The RRM controls access to the entry point objects that are used to access platform services. The resource manager will enforce the security policy for applications as defined by the respective SPs, limiting access to the platform resources as stipulated by the policy.

For each application (package), an Application Resource Manager (ARM) is introduced. This component will act as the authentication and resource allocation point. A client application will request a server application's ARM to enable the sharing of resources. The ARM will decide whether to grant the request based upon the client's credentials (associated privileges). At the time of application installation, the ARM also establishes a shareable interface connection with the platform, enabling the application to access methods that are essential for the application execution. The platform can access any method in the application context only after authorisation from the application's SP. The ARM also receives information regarding the requesting application. If the request is, from the system context, for a method that is not allowed to be accessed by the platform, then the ARM will indicate a security exception.

An Access Control List (ACL) is a private list and it is used to facilitate the implementation of hierarchical access mechanisms and privilege revocations. An ACL can be updated remotely by its corresponding SP (when the application connects with the SP's servers, the SP can update the ACL), changing the behaviour of its application's sharing mechanism. The ACL holds lists of granted permissions, received permissions (permissions to access other application's resources) and a cryptographic certificate revocation list of client applications. The structure of an ACL is under the sole discretion of its SP and it is stored as part of the ARM.

The operations of the firewall can be sub-divided into two distinctive phases. In phase one, a binding is established between the client and the server applications. This process includes authentication of the client's credentials and access privileges by the server's ARM. In the second phase, the client application requests resources in line with the privileges sanctioned by the ARM. In both these phases, the firewall mechanism facilitates individual authorised applications to accomplish the application sharing, while prohibiting unauthorised applications from accessing the resources of an application.

4.1.4 Decommissioning Process The decommissioning process involves deletion of all applications from a UCSC and removal of any user-specific data stored by the applications and card management system. The decommissioning process is initiated by the user in a manner similar to the ownership acquisition process (section 4.1.1). However, in the decommissioning process the user requests a UCSC to delete all applications in a manner similar to the one discussed in the previous section but this time the UCSC does not check for dependencies. Once all applications are deleted, the card security manager will delete the user-specific cryptographic keys (e.g. user signature key) and associated certificates. It will then request the deletion of ownership credentials that the user has set during the ownership acquisition process. After the decommissioning process is

completed, the UCSC reverts to the state it was in when the user acquired it from the card manufacturer (or UCSC suppliers). In other words, it is a blank UCSC.

5 Conclusion

The proposal for the UCOM began with a simple question "can a user have application control on a security-sensitive device like a smart card in a simple but secure manner?". The work on the UCOM has not yet resolved all the issues and modifications required to completely abandoned the ICOM. However, work to date has a common foundation namely "least interaction", which requires the user's involvement in different UCSC management operations to be kept to a minimum. This enabled us to design a secure, yet user friendly framework to support UCSC.

The work done up till now on the concept of UCOM has shown that a robust and secure system does not have to be difficult for ordinary users to understand/use. We consider that such effects, making the security of a system intuitive, seamless and requiring the minimum of user interaction, might lead the way for better, more reliable and secure systems.

References

1. P. Dusart, D. Sauveron, and K. Tai-Hoon, "Some Limits of Common Criteria Certification," *International Journal of Security and its Applications*, vol. 2, no. 4, pp. 11 – 20, October 2008.
2. D. Sauveron and P. Dusart, "Which Trust Can Be Expected of the Common Criteria Certification at End-User Level?" *Future Generation Communication and Networking*, vol. 2, pp. 423–428, 2007.
3. C. Xenakis and L. Merakos, "Security in Third Generation Mobile Networks," *Computer communications*, vol. 27, no. 7, pp. 638–650, 2004.
4. E. E. Schultz, "Research on Usability in Information Security," *Computer Fraud & Security*, vol. 2007, no. 6, pp. 8–10, 2007.
5. R. Anderson and T. Moore, "Information Security Economics–and Beyond," in *Advances in Cryptology-CRYPTO 2007*. Springer, 2007, pp. 68–91.
6. I. G. Askoxylakis, M. Pramateftakis, D. D. Kastanis, and A. P. Traganitis, "Integration of a Secure Mobile Payment System in a GSM/UMTS SIM Smart Card," in *Proceedings of the Fourth IASTED International Conference on Communication, Network and Information Security*, ser. CNIS '07. Anaheim, CA, USA: ACTA Press, 2007, pp. 40–50.
7. A. Whitten and J. D. Tygar, "Why Johnny Can'T Encrypt: A Usability Evaluation of PGP 5.0," in *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, ser. SSYM'99. CA, USA: USENIX Association, 1999, pp. 14–14.
8. *EMV 4.2*, Online, EMVCo Specification 4.2, May 2008. [Online]. Available: http://www.emvco.com/specifications.aspx?id=155
9. "Entity Authentication Assurance Framework," ITU-T, Geneva, Switzerland, Recommendation ITU-T X.1254, September 2012. [Online]. Available: http://www.itu.int/rec/T-REC-X.1254-201209-I

10. A. Mitrokotsa, Q. Z. Sheng, and Z. Maamar, "User-driven RFID applications and challenges," *Personal and Ubiquitous Computing*, vol. 16, no. 3, pp. 223–224, 2012.
11. 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)*, ser. LNCS, H. Y. YOUM and M. Yung, Eds., vol. 5932/2009. Busan, Korea: Springer, August 2009, pp. 20–35.
12. N. E. Petroulakis, I. G. Askoxylakis, and T. Tryfonas, "Life-logging in Smart Environments: Challenges and Security Threats," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5680–5684.
13. J. Laugesen and Y. Yuan, "What Factors Contributed to the Success of Apple's iPhone?" in *Proceedings of the 2010 Ninth International Conference on Mobile Business / 2010 Ninth Global Mobility Roundtable*, ser. ICMB-GMR '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 91–99.
14. "Near Field Communications (NFC). Simplifying and Expanding. Contactless Commerce, Connectivity, and Content," Online, ABI Research, Oyster Bay, NY, 2006. [Online]. Available: http://www.abiresearch.com/research/1000885-Near-Field_Communications_(NFC)
15. D. Sauveron, "Multiapplication Smart Card: Towards an Open Smart Card?" *Inf. Secur. Tech. Rep.*, vol. 14, no. 2, pp. 70–78, 2009.
16. "The GlobalPlatform Proposition for NFC Mobile: Secure Element Management and Messaging," GlobalPlatform, White Paper, April 2009.
17. "Mobile NFC Services," GSM Association, White Paper Version 1.0, 2007. [Online]. Available: http://www.gsmworld.com/documents/nfc_services_0207.pdf
18. 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)*, Bernady O. Apduhan, Osvaldo Gervasi, Andres Iglesias, D. Taniar, and M. Gavrilova, Eds. Fukuoka, Japan: IEEE Computer Society, March 2010, pp. 191–200.
19. "GlobalPlatform A New Model: The Consumer-Centric Model and How It Applies to the Mobile Ecosystem," GlobalPlatform, Whitepaper, March 2013.
20. P. Girard, "Which Security Policy for Multiplication Smart Cards?" in *Proceedings of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology*. Berkeley, CA, USA: USENIX Association, 1999, pp. 3–3. [Online]. Available: http://portal.acm.org/citation.cfm?id=1267115.1267118
21. S. Chaumette and D. Sauveron, "New Security Problems Raised by Open Multiapplication Smart Cards." *LaBRI, Université Bordeaux 1.*, pp. 1332–04, 2004.
22. R. N. Akram, K. Markantonakis, and K. Mayes, "A Dynamic and Ubiquitous Smart Card Security Assurance and Validation Mechanism," in *25th IFIP International Information Security Conference (SEC 2010)*, ser. IFIP AICT Series, Kai Rannenberg and V. Varadharajan, Eds. Brisbane, Australia: Springer, September 2010, pp. 161–171.
23. (Visited June, 2010) London Underground: Oyster Card. London Underground. United Kingdom. [Online]. Available: https://oyster.tfl.gov.uk/oyster/entry.do
24. (Visited December, 2010) Octopus. Octopus Holdings Ltd. Hong Kong, China. [Online]. Available: http://www.octopus.com.hk/home/en/index.html
25. R. N. Akram, K. Markantonakis, and K. Mayes, "Remote Attestation Mechanism based on Physical Unclonable Functions," in *The 2013 Workshop on RFID and IoT Security (RFIDsec'13 Asia)*, C. M. J. Zhou and J. Weng, Eds. Guangzhou, China: IOS Press., November 2013.

26. ——, "Remote Attestation Mechanism for User Centric Smart Cards using Pseudorandom Number Generators," in *5th International Conference on Information and Communications Security (ICICS 2013)*, S. Qing and J. Zhou, Eds. Beijing, China: Springer, November 2013.
27. J. Bringer, H. Chabanne, T. Kevenaar, and B. Kindarji, "Extending Match-On-Card to Local Biometric Identification," in *Biometric ID Management and Multimodal Communication*, ser. Lecture Notes in Computer Science, J. Fierrez, J. Ortega-Garcia, A. Esposito, A. Drygajlo, and M. Faundez-Zanuy, Eds. Springer Berlin / Heidelberg, 2009, vol. 5707, pp. 178–186, 10.1007/978-3-642-04391-8_23. [Online]. Available: http://www.springerlink.com/content/b16016708315549v/fulltext.pdf
28. R. N. Akram, K. Markantonakis, and K. Mayes, "A Privacy Preserving Application Acquisition Protocol," in *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-12)*, F. G. M. Geyong Min, Ed. Liverpool, United Kingdom: IEEE Computer Society, June 2012.
29. ——, "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.
30. ——, "Coopetitive Architecture to Support a Dynamic and Scalable NFC based Mobile Services Architecture," in *The 2012 International Conference on Information and Communications Security (ICICS 2012)*, K. Chow and L. C. Hui, Eds. Hong Kong, China: Springer, October 2012.
31. D. A. Basin, S. Friedrich, J. Posegga, and H. Vogt, "Java Bytecode Verification by Model Checking," in *CAV '99: Proceedings of the 11th International Conference on Computer Aided Verification*. London, UK: Springer-Verlag, 1999, pp. 491–494.
32. *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*, Sun Microsystem Inc Std. Version 3.0.1, May 2009.
33. D. A. Basin, S. Friedrich, and M. Gawkowski, "Verified Bytecode Model Checkers," in *TPHOLs '02: Proceedings of the 15th International Conference on Theorem Proving in Higher Order Logics*. London, UK: Springer-Verlag, 2002, pp. 47–66.
34. R. N. Akram, K. Markantonakis, and K. Mayes, "Firewall Mechanism in a User Centric Smart Card Ownership Model," in *Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010*, ser. LNCS, D. Gollmann, J.-L. Lanet, and J. Iguchi-Cartigny, Eds., vol. 6035/2010. Passau, Germany: Springer, April 2010, pp. 118–132.
35. ——, "Application-Binding Protocol in the User Centric Smart Card Ownership Model," 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.