Trusted Platform Module for Smart Cards

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

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

***Abstract*—Near Field Communication (NFC)-based mobile phone services offer a lifeline to the under-appreciated multi-application smart card initiative. The initiative could effectively replace heavy wallets full of smart cards for mundane tasks. However, the issue of the deployment model still lingers on. Possible approaches include, but are not restricted to, the User Centric Smart card Ownership Model (UCOM), GlobalPlatform Consumer Centric Model, and Trusted Service Manager (TSM). In addition, multiapplication smart card architecture can be a GlobalPlatform Trusted Execution Environment (TEE) and/or User Centric Tamper-Resistant Device (UCTD), which provide cross-device security and privacy preservation platforms to their users. In the multiapplication smart card environment, there might not be a prior off-card trusted relationship between a smart card and an application provider. Therefore, as a possible solution to overcome the absence of prior trusted relationships, this paper proposes the concept of Trusted Platform Module (TPM) for smart cards (embedded devices) that can act as a point of reference for establishing the necessary trust between the device and an application provider, and among applications.**

***Keywords*-Smart Card, Near Field Communication, Trusted Execution Environment, GlobalPlatform Consumer Centric Model, Trusted Platform Module, User Centric Smart Cards.**

I. Introduction

The multi-application smart card initiative has gained momentum due to the introduction of Near Field Communication (NFC) [1]. NFC enables a mobile phone to emulate a contactless smart card [2, 3]. Applications installed on the secure element[1] [4] of an NFC-enabled mobile phone could utilise this functionality to communicate with external entities through the contactless interface [5]. Hereafter, the terms "smart card" and "secure element" are used interchangeably.

NFC field trials are either adopting the traditional card issuer centric approach, termed the Issuer Centric Smart Card Ownership Model (ICOM) or a refined form of it known as Trusted Service Manager (TSM) [6]. In these trials, different stakeholders are trying to leverage this new technology in their favour [5, 7]–[9]. In addition, there is a debate around the issue that ICOM has already decelerated the multiapplication smart card initiative [10], and why should this model be tried again. There are some positive points for using the ICOM as it has proven security, operational and service architecture.

Along with the ICOM, several other smart card ownership models are proposed that include User Centric Smart Card Ownership Model (UCOM) [10] and the GlobalPlatform Consumer-Centric Model [11]. In addition, multiapplication smart card technology can be scaled up to provide a general purpose security and privacy preservation platform in different computing environments (i.e. mobiles, tablets, and personal computers). Examples of such initiatives are the GlobalPlatform Trusted Execution Environment (TEE) [12] and User Centric Tamper-Resistant Device (UCTD) [13].

Multiapplication smart cards, under any proposal, will have applications from diverse application providers. In addition, smart cards might install applications over the internet. In such an environment:

1) How is the application provider going to establish trust in the smart cards?
2) How can the applications installed on the smart card trust each other?
3) How can the smart card platform verify the state of the application at installation and during its execution?

This paper will look at the mechanisms to remotely establish trust in a smart card platform and applications. In addition, it will examine how a Trusted Platform Module (TPM)-based architecture can fill the gaps and provide a service that answers the questions above.

A. Structure of the Paper

Section II, discusses the concept of the Trusted Computing Base (TCB) for smart cards. The discussion of need-of-trust in the future multiapplication smart card based technologies is extended in section III and the architecture of the proposed TCB for smart cards is also addressed. Finally, future research directions and conclusions are presented in section IV.

II. Trusted Computing Base for Smart Cards

This section discusses the motivation behind proposing a Trusted Computing Base (TCB) for smart cards and how it fits into the overall smart card services architecture.

A. Motivation

The notions of trust and trustworthiness are fundamental to the field of computer security, but the definitions are

[1]Secure Element: A secure electronic device that can store data and execute programmes. Examples include Universal Integrated Circuit Cards (UICC), Embedded Secure Elements and Secure Memory Cards.

ambiguous. Trust and trustworthiness mean different things in different computing environments and in different contexts (e.g. patient records, consumer devices and online banking). Therefore, in the context of this paper they are defined by paraphrasing the definition of trusted computing from [14, 15]. The term "trusted" refers to the level of assurance and validity that an entity has regarding the security and operational reliability of a multiapplication smart card (i.e. both software and hardware). The term "trustworthy" means the assurance gained by the entity after using specified verification mechanisms which ensure that the smart card behaves in a similar fashion to that stated in the provided assurance (i.e. Common Criteria Evaluation Assurance Level [16]).

In the ICOM, trust is established based on ownership. As smart cards are owned by their respective card issuers, so they would define and enforce the security and functional policies of the final product. The respective smart card manufacturers would build the product to match the card issuer's requirements. If required, as in the case of banking cards, the card manufacturer will have their smart cards certified by an independent evaluation authority like Common Criteria (CC) [16]. The evaluation of the smart cards establishes trust in the product. After evaluation of the smart cards, generally there is no implemented mechanism that validates whether their state is as it was at the time of evaluation.

From the application provider's point of view, in the ICOM they have to trust the card issuer, which implicitly establishes a trust in respective smart cards. Similar arguments can also be applied in reverse; a card issuer trusts an application because of the off-card relationship with the application provider. In such a closed environment with a fixed number of applications on the smart card, the notion of trust and its dynamic validation might not be necessary.

The GlobalPlatform Consumer-Centric Model (GP-CCM) and the UCOM both push for the empowerment of the smart card user. A card user (cardholder) may have the privilege of deciding which applications (s)he wants to install or delete. The term "control" in the UCOM and GP-CCM context means the freedom of "choice" given to cardholders to install or delete any application from their smart cards without any restriction from the card manufacturers, card issuers or TSM. The card issuers and/or application providers are not required to provide a smart card to their customers. Instead, they develop their application and make it available to their authorised user. The application can be downloaded to those smart cards that abide by the terms and conditions of the application providers. In the UCOM, the application provider is referred as a Service Provider (SP). An SP defines an Application Lease Policy (ALP) [17] that governs the lease of its applications to authorised customers. The ALP stipulates the security and operational requirements of an application and the host smart card platform has to ensure that it provides adequate functionality to meet these requirements [18].

A smart card is acquired by cardholders directly from a card provider, which can be a card manufacturer, an SP or a third party. SPs of the respective applications installed by a cardholder might not have any off-card/prior trust relationship with the respective card provider. Therefore, unlike the ICOM, neither smart cards nor applications have a chain of trust through their issuing authorities and owners, respectively. When a cardholder requests installation of an application from its SP, the SP would like to have a dynamic assurance, which gives it the trustworthiness of the host platform, and vice versa [18]. To establish and evaluate the trustworthiness of a platform or an application requires a trust base. To provide a trusted base for the UCOM and GP-CCM initiative, this paper proposes a trusted entity on the smart card, known as the Trusted Execution and Environment Manager (TEM). Such an entity would play a crucial role in establishing the trustworthiness of smart cards [18], platform assurance [19], smart card firewall mechanisms [20], trusted execution environments and smart card content backup/restoration mechanisms [21]. From the point of view of different stake-holders in various smart card models, a TEM should be able to provide, at a minimum, the following services.

1) Confidence in Current State: Provide assurance and validation that the state of a smart card (software and hardware) is as secure as it was at the time of CC evaluation.
2) Trust in the Downloaded Application: Ensure that the application is downloaded and personalised in a secure and reliable fashion. Provide proof to the appropriate SP that there was no modification of the application during the download and installation process.
3) Secure State and Application Sharing: Provide assurance and validation services which an application can use to validate its current state, and verify the state of other application(s) with which it establishes application sharing.
4) Secure Execution: Provide a trusted execution environment which ensures that an application is executed in a trusted and secure environment.
5) Simulator Detection: Provide verification and validation mechanisms to ascertain both the existence of a smart card and that the item is not a smart card simulator (hardware genuineness [22]).

B. Related Work

At the time of writing, there appears to be no related work that discusses the need for or design of a TCB for smart cards. Nevertheless, the TPM architecture for embedded devices was proposed by Araja et al. in [23]. Kurt et al. discussed the possibility of using smart cards as TPMs for mobile phones [24], based on their earlier work in which they proposed an architecture to implement the Mobile Trusted Module (MTM) on Java Card [25]. Furthermore, smart cards are used with TPMs to provide a secure and reliable architecture [26]. In relation to NFC, TPM and MTM have been proposed by [27, 28]. In the literature there are countless examples of using smart cards in relation to TPMs and the above-mentioned architectures are by no means an exhaustive list, or the best possible material.

III. Trusted Environment & Execution Manager (TEM)

This section discusses the architecture of a TEM specific for smart cards, and highlights how the TEM differs from a typical TPM not only in architectural but also in operational context.

A. Architecture

The overall architecture of a UCOM based smart card is illustrated in Figure 1. The TEM is illustrated as a layer between the smart card hardware and the runtime environment. This illustration provides a semantic view of the architecture and does not imply that all communication between the runtime environment and the hardware goes through the TEM.

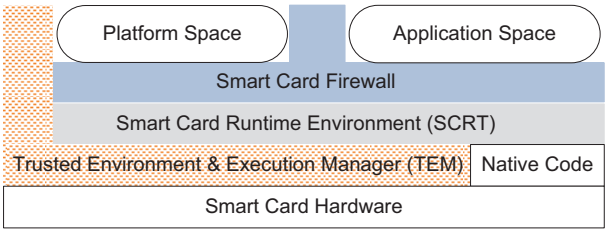

Figure 1. Smart Card Architecture in with TEM

If general TPM requirements are analysed [15], the basic building blocks in the hardware required to build a TPM chip are already available on smart cards. Therefore, most of the functionality of the TEM would be implemented in software and it would not impose any additional hardware requirement on the host platform. The detailed TEM architecture is shown in Figure 2.

Figure 2 depicts native code and smart card hardware as complementary components of the TEM. This is because the TEM does not need separate hardware for its operations. It will utilise the existing services provided by the smart card hardware. To avoid duplicating the code, the TEM uses the native code implementation of cryptographic services like encryption/decryption, digital signature and random number generation.

1) Interface: The interface manages communication with external entities that can either be on-card or off-card entities. Any request that the interface receives is interpreted: if it is a well-formed request and the requesting entity is authorised to do so, then the interface will redirect the request to the intended module in the TEM. The interface during the interpretation of the request will enforce the access policy of the TEM as defined by the access control module (discussed in section III-A3). To manage these relationships with the authorised entities, the TEM should have a mechanism to establish the relationship in the first place. Therefore, at the time of installation of an application, a binding (symmetric key) is generated between the downloaded application and the TEM. For all subsequent communications, the application would use this key when requesting the TEM [29]. The protocol that establishes this binding is managed by the interface and the binding is stored in the key/certificate manager (section III-A4) and corresponding access privilege in access control module.

2) Attestation Handler: During the application installation process, both an application and a smart card platform would need to verify each other's current state to gain assurance of their trustworthiness. An application can only request attestation for either itself or the respective platform. It cannot request attestation for other applications on the smart card concerned. However, to facilitate the application sharing mechanism [20] an application can issue an authorisation token. The attestation handler will then provide the attestation of the token-issuing application to the requesting application [30].

3) Access Control: At the time of application installation, the SP involved would request attestation of the card platform. However, no information regarding any of the other applications installed on the card would be provided to the SP at this stage. Once the application is installed, it can request attestation only for itself and not for any other applications. These restrictions are required to avoid privacy issues like application scanning attacks [30].

4) Key & Certificate Manager: The key & certificate manager manages the keys and certificates that a TEM stores in the non-volatile memory (EEPROM [32]). Contrary to the general TPM architecture, there are no migratable keys in the TEM. The TEM signature key pair and certificate is the permanent key and certificate (it can be considered as the endorsement key in the general TPM architecture). Besides managing the keys and certificates, it also generates them. Therefore, it is a combination of key generation and non-volatile memory components of the general TPM.

The key & certificate manager stores the evaluation certificates which are provided by the respective applications. Therefore, when an application requests attestation, the TEM does not return the hash value of the application. In fact, it returns an evaluation of whether the current state complies with the state for which the evaluation certificate was issued. Therefore, the decision whether an application is trustworthy or not is actually made by the TEM. If the evaluation fails, then depending upon the application or platform policy it might either block the application or delete it (and inform the cardholder and respective SP).

5) Ownership Manager: This component manages the ownership of a smart card. When a smart card is acquired by a user either from a card manufacturer or a card supplier, it is under the default ownership of the card manufacturer/supplier. The user then initiates the ownership acquisition process that requires the user to provide personal information (i.e. name and date of birth) and their Card Management Personal Identification Number (CM-PIN). The TEM will then generate a signature key pair specific to the cardholder along with a certificate that will also include the user information. Although this key is assigned to the cardholder, it will be protected by the TEM.

6) TSM Scheme Registration Manager: This module is optional and it facilitates Competitive Architecture for Smart Cards. For further details please refer to Akram et al [31].

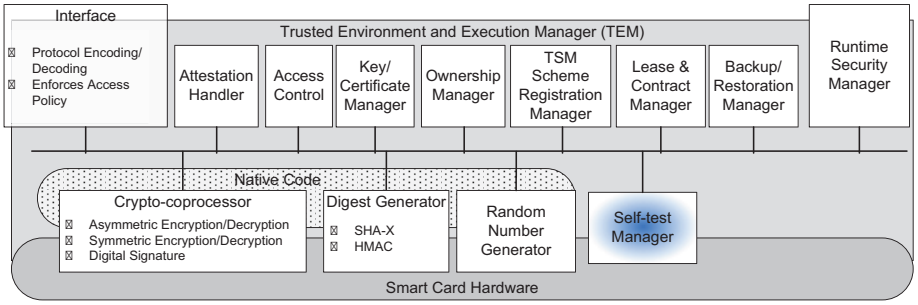

Figure 2. Trusted Platform Module for Smart Card Architecture

7) Lease & Contract Manager: An SP would lease its application to a smart card (cardholder) and the card would assure that it would abide by the SP's ALP. The lease contract is signed by the TEM with the user's signature key and as these keys are stored/restrict access only to the TEM, the signing and storage of the contracts are on the TEM. The cardholder can retrieve these contracts after providing the CM-PIN if he/she needs to. Similarly, individual applications can also retrieve their own contracts from the TEM repository.

8) Backup/Restoration Manager: A cardholder may download multiple applications onto her smart card. If she loses her smart card, she will lose access to all of the applications (and related services). One possible approach can be to acquire a new card and then manually install all the applications again. However, another approach could be that a user creates a backup of the installed applications and restores the backup to a new smart card, if required. This backup mechanism is credential-based (a token issued by the SPs and not the actual application) and it is stored securely at a remote location [21]. When users lose their smart cards, they only need to get a new smart card and then initiate the restoration process, which will take each credential from the backup and initiate the application download process with respective SPs. The restoration process can also request the respective SPs to block (revoke the lease) their application(s) installed on the stolen/lost device.

9) Self-test Manager: For security validation, the TEM implements a validation mechanism that is divided into two parts: tamper-evidence and reliability assurance. Smart cards are required to be tamper-resistant devices [32] and for this purpose card manufacturers implement hardware-based tamper protections. The tamper-evidence process verifies whether the implemented tamper-resistant mechanisms are still in place and effective. The reliability assurance process, on the other hand, verifies that the software part of the smart card that is crucial for its security and reliability has not been tampered with.

A TEM tamper-evidence process should provide the properties listed below:

1) Robustness: On input of certain data, it always produces the associated output.
2) Independence: When the same data is input to a tamper-evidence process on two different devices, it outputs different values.
3) Pseudo-randomness: The generated output should be computationally difficult to distinguish from a pseudo-random function.
4) Tamper-evidence: An invasive attack to access the function should cause irreversible changes, which render the device unusable.
5) Assurance: The function can provide assurance (either implicitly or explicitly) to independent (non-related) verifiers. It should not require an active connection with the device manufacturer to provide the assurance. The assurance refers to the current hardware and software state as it was at the time of third party evaluation.

For the TEM tamper-evidence process there are several candidates including: active (intelligent) shield/mesh [32]; Known Answer Test (KAT) [33], hard-wired HMAC key, attestation based on PRNG [22]; and Physically Unclonable Function (PUF) [34]. Two algorithms that provide tamper-evidence and reliability based upon PUF and PRNG based validation mechanisms are discussed in [35] and [36], respectively.

10) Runtime Security Manager: The purpose of the runtime security manager is to enforce the security counter-measures defined by the respective platform. To enforce the security counter-measures, the runtime security manager has access to the heap area (e.g. method area, Java stacks) and can be implemented as either a serial or a parallel mode.

A serial runtime security manager will rely on the execution engine of the Java Card Virtual Machine (JCVM) [37] to perform the required tasks. This means that when an execution engine encounters instructions that require an enforcement of the security policy, it will invoke the runtime security manager that will then perform the checks. If successful the execution engine continues with execution, otherwise, it will terminate. A parallel runtime security manager will have its own dedicated hardware (i.e. processor) support that enables it to perform checks simultaneously while the execution engine is executing an application. Having multiple processors on a smart card is technically possible [32]. The main question regarding the choice is not the hardware, but the balance between performance and latency. Therefore, theoretically we can assume that a serial runtime security manager will have low performance but also a low latency value, while a parallel runtime security manager will have a good performance

measure but a higher latency value.

It is obvious that implementation of additional components like runtime security managers will also incur additional economic costs (i.e. increase in the price of a UCTD). The security measures that could be enforced are: 1) Operand Stack Integrity 2) Control Flow Analysis 3) Bytecode Integrity

IV. Conclusion and Future Research Directions

This paper has discussed the importance of having a TCB on embedded devices like smart cards. The paper discussed existing smart card architectures and illustrated how they fall short of a TCB framework, then proposed the TEM architecture and discussed its different components. Finally, it described a few of the applications that will benefit from such an architecture. Future research involves implementing the TEM as part of the Java Card architecture and doing some performance and reliability evaluation.

References

[1] "Near Field Communications (NFC). Simplifying and Expanding. Contactless Commerce, Connectivity, and Content," ABI Research, Oyster Bay, NY, 2006.

[2] *ISO/IEC 14443-1: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards,*, International Organization for Standardization (ISO) Std., Rev. 2nd Edition, June 2008.

[3] J. J. Park, S.-J. Wang, and D. Sauveron, "Security technologies and applications for convergence environments," *Security and Communication Networks*, vol. 5, no. 3, pp. 249–251, 2012.

[4] D. D. Kastanis and I. G. Askoxylakis and A. Traganitis,"An efficient authentication scheme for contactless smartcards using elliptic curve cryptography," *Communication, Network, and Information Security*, 2006, 28–33

[5] "The GlobalPlatform Proposition for NFC Mobile: Secure Element Management and Messaging," GlobalPlatform, WP, April 2009.

[6] "Mobile NFC Services," GSM Association, White Paper v1.0, 2007.

[7] "Pay-Buy-Mobile: Business Opportunity Analysis," GSM Association, White Paper 1.0, November 2007.

[8] 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", *System*, 2007, volumn 12, p 13

[9] Q. Z. Sheng, S. Zeadally, A. Mitrokotsa, and Z. Maamar, "RFID technology, systems, and applications," *Journal of network and computer applications*, vol. 34, no. 3, pp. 797–798, 2011.

[10] 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 CS, March 2010.

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

[12] "GlobalPlatform Device Technology: TEE System Architecture," GlobalPlatform, Specification Version 0.4, October 2011.

[13] R. N. Akram, K. Markantonakis, and K. Mayes, "User Centric Security Model for Tamper-Resistant Devices," in *the 8th IEEE International Conference on e-Business Engineering (ICEBE 2011)*, J. Li and J.-Y. Chung, Eds. Beijing, China: IEEE Computer Science, October 2011.

[14] E. Gallery and C. J. Mitchell, "Foundations of security analysis and design iv," A. Aldini and R. Gorrieri, Eds. Berlin, Heidelberg: Springer-Verlag, 2007, ch. Trusted mobile platforms, pp. 282–323.

[15] "Trusted Computing Group, TCG Specification Architecture Overview," The TCG, Oregon, USA, rev 1.4, August 2007.

[16] *Common Criteria for Information Technology Security Evaluation*, Common Criteria Std. Version 3.1, August 2006.

[17] 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.

[18] ——, "A Dynamic and Ubiquitous Smart Card Security Assurance and Validation Mechanism," in *25th IFIP International Information Security Conference (SEC 2010)*, ser. IFIP AICT Series, K. Rannenberg and V. Varadharajan, Eds. Australia: Springer, September 2010.

[19] ——, "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.

[20] ——, "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*, D. Gollmann, J.-L. Lanet, and J. Iguchi-Cartigny, Eds., vol. 6035/2010. Passau, Germany: Springer, April 2010, pp. 118–132.

[21] ——, "Recovering from Lost Digital Wallet," in *The 4th IEEE International Symposium on Trust, Security, and Privacy for Emerging Applications (TSP-13)*, F. G. M. Y. Xiang and S. Ruj, Eds. Zhangjiajie, China: IEEE CS, November 2013.

[22] R. Kennell and L. H. Jamieson, "Establishing the genuinity of remote computer systems," in *Proceedings of the 12th conference on USENIX Security Symposium - Volume 12*. CA, USA: USENIX, 2003.

[23] N. Aaraj, A. Raghunathan, and N. K. Jha, "Analysis and design of a hardware/software trusted platform module for embedded systems," *ACM Trans. Embed. Comput. Syst.*, vol. 8, pp. 8:1–8:31, January 2009.

[24] K. Dietrich and J. Winter, "Implementation Aspects of Mobile and Embedded Trusted Computing," in *Trust '09: Proceedings of the 2nd International Conference on Trusted Computing*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 29–44.

[25] K. Dietrich, "An integrated architecture for trusted computing for java enabled embedded devices," in *STC '07: Proceedings of the 2007 ACM workshop on Scalable trusted computing*. NY, USA: ACM, 2007.

[26] A. Klenk, H. Kinkelin, C. Eunicke, and G. Carle, "Preventing identity theft with electronic identity cards and the trusted platform module," in *Proceedings of the Second European Workshop on System Security*, ser. EUROSEC '09. New York, NY, USA: ACM, 2009, pp. 44–51.

[27] G. Madlmayr, "A mobile trusted computing architecture for a near field communication ecosystem," in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, ser. iiWAS '08. NY, USA: ACM, 2008, pp. 563–566.

[28] R. Toegl, "Tagging the Turtle: Local Attestation for Kiosk Computing," in *ISA '09: Proceedings of the 3rd International Conference and Workshops on Advances in Information Security and Assurance*. Berlin, Heidelberg: Springer, 2009, pp. 60–69.

[29] R. N. Akram, K. Markantonakis, and K. Mayes, "Cross-Platform Application Sharing Mechanism," in *10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-11)*, H. Wang, S. R. Tate, and Y. Xiang, Eds. Changsha, China: IEEE Computer Society, November 2011.

[30] ——, "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.

[31] ——, "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 CS, November 2013.

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

[33] *FIPS 140-2: Security Requirements for Cryptographic Modules*, Online, National Institute of Standards and Technology (NIST) FIPS Publication, Rev. Supersedes FIPS PUB 140-1, May 2005.

[34] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon Physical Random Functions," in *Proceedings of the 9th ACM conference on Computer and communications security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 148–160.

[35] 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.

[36] ——, "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.

[37] *Java Card Platform Specification*, Sun Microsystem Inc Std. Version 3.0.1, May 2009.