Secure Deployment of Applications to Fielded Devices and Smart Cards

William G. Sirett, John A. MacDonald, Keith Mayes and Konstantinos Markantonakis

Smart Card Centre – Information Security Group
Royal Holloway, University of London, UK

Abstract. This work presents a process of deploying applications securely to fielded devices with smart cards whilst taking into consideration the possibility that the client device could be malicious. Advantages of the proposed process include; caching functionality upon the device, optimal use of resources, employment of nested security contexts whilst addressing fielded infrastructures and a homogeneous solution. This work outlines a targeted scenario, details existing malicious device activity and defines an attacker profile. Assumptions and requirements are drawn and analysis of the proposal and attack scenarios is conducted. Advantages and deployment scenarios are presented with an implementation the process using Java and specific standards.

1 Introduction

Consider the situation of having a large field base of equipment which interact with secure elements such as smart cards. Examples of these infrastructures are 3GPP mobile network [1] and satellite TV Set-Top-Boxs (STB) [2, 3]. Functionality can be split across two applications; *fielded host device* (referred to as device) application and smart card application. This can effectively use the resources available. For example, the device's greater storage capacity and processing power or the smart card's tamper resistant qualities [4, 5]. New applications sometimes need to be distributed in response to emerging security problems or the deployment of new functionality. The field base could be recalled or new smart cards issued but the costs could be prohibitive. Replacing the circa 260m GSM cellular phones in Western Europe and accompanying Subscriber Identity Module (SIM) cards would be an expensive undertaking.

New device and smart card applications need to be securely deployed to remotely fielded devices whilst maintaining integrity, confidentiality and authenticity. This work demonstrates, using specific technologies, how to securely deploy a device application that is capable of securely installing a smart card application in the field. The device and smart card applications are separated and each maintains individual security contexts maintaining integrity and confidentiality. As device functionality increases so does the scope for user modification and subversion of security measures, therefore, the device can be considered less trusted than the smart card [6]. The approach uses a homogeneous solution to utilise the device as a high capacity cache whilst protecting the smart card. Our solution considers resource use in a constrained environment by maintaining

G. Sirett W., A. MacDonald J., Mayes K. and Markantonakis K. (2006).
Secure Deployment of Applications to Fielded Devices and Smart Cards.
In *Proceedings of the 4th International Workshop on Security in Information Systems*, pages 195-206

two security contexts to separate device from smart cards and not only provides a secure method of deploying smart card applications but couples of device and smart card applications.

The structure of the work is as follows; the architecture of the problem area is introduced and assumptions and requirements drawn against it. Existing approaches are presented and evaluation is conducted concerning their suitability to the problem area. Our own solution is introduced; including architecture, overview and detailed exploration of process. The advantages of this solution are presented followed by examples of industrial implementation and possible scenarios of deployment. Finally, concluding remarks are provided on the work conducted and future research direction.

2 Scenario Specification

In this section we graphically represent the targeted scenario, explore the possible malicious behaviour of the user device and outline an attacker profile alongside assumptions and requirements for a solution.

2.1 Targeted Scenario

Fig 1 shows the entities involved; server, device and smart card. Although this paper focuses upon smart card technology the smart card entity could be replaced by other technologies that offer tamper resistant abilities and a programmable operating system such as a USB token or dongle. Each entity in Fig 1 is shown holding one application but are capable of storing multiple applications.

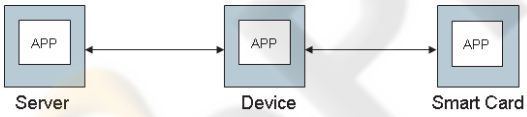

Fig. 1. Scenario Architecture.

The presence of the secure entity, or smart card, in this system implies the need for a tamper resistant device at the client site – this is due to the possible malicious or subversive activities of the remote user. In this targeted scenario the remote user is consuming some type of service. The network provider must provide legitimate clients across a hostile network to a user that cannot be assumed to be trusted [7]. The network/service provider must place trust in the security mechanisms present at the client site to protect its interests. However, the user has full control and access to both hardware and software on the remote device and that unlimited amount of time and resources to attack the system [7]. The identified attacker is capable of reading, modifying, creating or deleting any communication between the device and smart card.

The most typical attacker considered by this work is the malicious user, educated by Internet communities, that is seeking a simple method of circumnavigating security measures on their mobile device. It is common for remote users of a system to attempt to break security measures; common examples demonstrated in the satellite television [8],

digital music and mobile telecommunication industries [9]. Smart card are susceptible to side channel attacks but this attack is beyond the scope of the standard user and is often only a plausible option for security labs or highly funded organisations. However, it is worth noting the falling costs and increased automation of side channel attacks and this will make them a more frequently employed tool [10–14].

2.2 Assumptions

The definition of assumptions allow us to eliminate issues that have been addressed elsewhere and to focus upon key problems. The first assumption (A1) states that the smart card or security entity employed is tamper resistant. No element is *tamper proof* [4] but a degree of *tamper resistance* can be attributed to a smart card. It is assumed that a secret securely placed into a smart card remains private. Assumption 2 (A2) refers to an existing shared secret between card and server; this not only infers a shared symmetric key but also any equivalent Public Key Infrastructure (PKI) mechanisms. Assumption 3 (A3) states that the server is capable of placing a key or certificate upon the device. This is possible in some existing architectures; for example the Satellite TV industry has the ability to alter fielded smart cards [15] and the mobile industry has similar functionality via the GSM standard and the 03.48 mechanism [16]. These mechanisms are looked at in Section 3 and have some limiting features. Assumption 4 (A4) states that all entities can handle multiple applications; this is fairly straightforward but addresses any ambiguity in the capability of the smart card entity. This assumption allows the consideration of handling multiple device and smart card applications within the architecture and validates the aims of this work. Assumption 5 (A5) assumes the smart card is capable but unprepared, this is to define that the card is compatible with all standards that the solution may require. In practice not all smart cards in the field will have the storage, processing or adherence to industry standards required. The final assumption (A6) declares that the device is open to malicious interference and considered less trusted than the smart card:

(A1) Smart card is tamper resistant,
(A2) Secret exists between smart card and server,
(A3) Server can place certificate and keys onto smart card securely,
(A4) All entities can handle multiple applications,
(A5) Smart card is considered capable but unprepared,
(A6) Device could be malicious and considered less trusted than smart card.

2.3 Requirements

The first requirement (R1) demands that the secure deployment of device and accompanying smart card applications is possible with a solution. R2 takes this further in stating that the solution arrived at must be applicable to fielded equipment. Fielded equipment refers to a remote host device and smart card in the hands of the user. The server is inclined to place a greater degree of trust in the smart card than the device; as the smart card has tamper resistant qualities (A1) and was issued by a trusted source. The level of trust that exists between the smart card and server must not be afforded to the device

(R3). The literature [17, 8] has examples of the possible malicious nature of devices in the field and it is a reasonable requirement to keep it one step removed from the server smart card security context. The smart card applications must be kept confidential and arrive on-card with integrity intact (R4, R5). The smart card is a secure environment and anything arriving in it must be handled securely prior to and during installation. These requirements also place constraints on what the device can do whilst handling the smart card application and must be cryptographically enforced. A successful solution will minimise demands on the resources of the system where possible (R6) and be homogeneous (R7) in nature. This will simplify the solution as procedures will be optimised for the constrained environment of a smart card [9] and the mechanisms used will interact in a uniform manner:

(R1) Secure deployment of device and smart card applications,
(R2) Applicability to fielded equipment,
(R3) Device not to be afforded level of trust that exists between smart card and server,
(R4) Confidentiality assured of smart card applications,
(R5) Integrity assured of smart card applications,
(R6) Optimised resource usage,
(R7) Homogeneous solution.

3 Existing Approaches

Several industries address individual problems and these are discussed in this section. The ETSI TS03.48 [16] provides end to end security for any Short Message Service (SMS) going to and from a SIM to network operator [18, 19]. However, the payload and performance are limited and applications are restricted [20]. It also only addresses the mobile industry needs and a broader solution is required. A secure device application installation procedure is defined using J2ME [21, 22] with the MIDP2.0 profile [23]. The GlobalPlatform Card Specification [24] defines a card application installation protocol and the GlobalPlatform Device API V2.0 [25] could be used to implement a bespoke solution.

Finally, a pragmatic approach remains; equipment could be recalled and updated in a trusted environment or new equipment deployed. Applying this to large field bases of devices could prove prohibitively expensive in terms of time and costs in comparison to a deployable software solution. No overall homogeneous solution is defined for our targeted scenario (section 2.1) but some individual aspects are provided for. The standards or protocols that exist to securely deploy applications to fielded devices are limited; either in industry applicability, bandwidth or performance. However, some of the specifications will prove useful in an implementation or proof of concept model but further enhancements are required.

4 Proposed Solution

This section introduces the solution architecture and fully explores both the process and advantages.

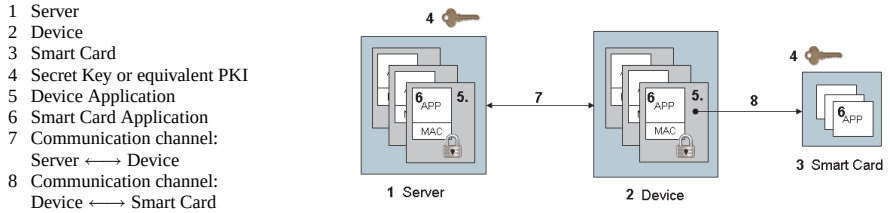

Fig. 2. Architecture of Solution definition.

4.1 Architecture

Fig 2 shows the architectural solution[1]. The system shows multiple applications running on each entity and an overall residing secret between server and smart card. The device is the only entity that can communicate with the card in the field and has been shown to be susceptible to malicious modification.

4.2 Process

(P1) *Preparation of Smart Card Application*:
The server prepares the smart card application using the shared secret between the card and server. It encrypts and hashes the application code knowing that the smart card can confirm origin and confidentiality. The server must know the smart card's unique identifier to be able to ascertain the corresponding secret to be used. This is achieved either through an initial authentication of the card or the knowledge is already known via another means.

(P2) *Preparation of Device Application*:
The server prepares the device application; the first stage is to embed the secure smart card application into the body of the device application. The encrypted smart card byte code becomes an integral part of the device application. The server then, using whatever security context that exists between itself and the device, signs the prepared device application.

(P3) *Server to Device Transfer*:
The Server uploads the device application with its embedded encrypted smart card application to the device. This process can be initiated by the device or the server can request the upload or a whole host of prepared device applications could be pre-installed on the device before deployment. However, this third option is in violation of R2 as it does not addressed the fielded device scenario.

(P4) *Device Application Installation*:
The device runtime environment, using the security context between itself and the server, authenticates the device application before installation. This confirms a legitimate server sent the application and that the code has its integrity intact. As a byproduct the embedded bytes of the smart card application are also verified as it is an integral part of the device application code. The device does not have possession of the secret used to protect the smart card application and therefore can

[1] Please note that the label MAC refers to Message Authentication Code.

only securely store the encrypted information for later use. It stores the smart card application whilst having its own access rights restricted.

(P5) *Smart Card Installation Initiation*:
As it stands the device has an application installed and the protected smart card application stored locally. The card has not installed anything up to this point. When required either by the server or user direction the device can initiate smart card application installation.

(P6) *Server Authentication*[2]:
As the smart card is a secure entity and only trusts the server, the server's authentication must first be received before any installation can take place. Only once this is received will the card continue the process.

(P7) *Checking of Smart Card Application*:
The shared secret between smart card and device allows for the on-card verification of the smart card application. Once confirmed the encrypted bytes are decoded on-card and installed successfully. The architecture now has new device and smart card applications installed whilst maintaining given security requirements.

4.3 Advantages

This solution provides a caching function which allows smart card applications to be installed, when required, from the device without downloading from the server. For example, a device could have multiple applications available for use by the user and the smart card element is only installed when required. It could be that the application is seldom used or that storage constraints upon the smart card are called into play. Additionally, communication from the server is optimised as only one download is required to deliver two applications to the client side. To install a new smart card application from the device, the server need only perform one authentication. The device has much greater capacity compared to the smart card in terms of storage and could hold many device applications (and therefore the embedded smart card applications). This allows the device and smart card to provide a greater amount of functionality as not all applications will need to be on the card at any given time. The storage strengths of the device are used to fully exploit the capacity of the smart card.

The device becomes a secure holding area as the smart card application is protected using a shared secret known only to smart card and server or PKI mechanism. Two combined layers of security allow the compartmentalisation of the security requirements of the different entities. The device is trusted to store and deliver the smart card application but does not install its own applications or read/modify the smart card applications. The card can independently confirm the integrity of any application before installation. Furthermore, this solution can be implemented on fielded devices and negates the need for device recall or replacement.

Finally, the combination of applications into one entity provides a much simpler solution than some alternatives that use separate download events and, sometimes, mechanisms. For example, the mobile network may employ GSM 03.48 to securely install a

[2] It should be noted that the necessity of this stage is not ideal and off-line authentication would present greater value. This could not be achieved with the available standards at present.

smart card application onto a SIM using SMS messages whilst downloading the device application via a higher bandwidth channel. The server is under additional pressure in terms of communication overheads and complexity.

5 Implementation

This section proposes an industrial implementation that utilises a range of technologies, standards and practises to demonstrate the functionality of the proposed solution. This section introduces the technologies involved and then goes onto discuss the implementation of the process detailed previously.

5.1 Platform

The platform chosen was the Java, which is pervasive in the marketplace [26] and the technology and tools are widely available. The smart card used supports the Java Card standards [27, 28] and GlobalPlatform 2.1.1 [24]. The device has a J2ME Java Runtime Environment (JRE) and supports two enabling specifications; JSR-118: Mobile Information Device Profile 2.0 [23] and JSR-177: Security and Trust Services API for J2ME [29]. JSR-118 extends the device profile and provides key security functionality [30]. Whereas, the JSR-177 provides J2ME applications with additional APIs which includes the provision of cryptographic functionality and the ability to communicate with a security element using the ISO standards [31, 32].

5.2 Process Implementation

This section shows how to realise the proposed solution in Section 4 (S=Server, D=Device and C=Smart Card).

(P1) *Preparation of Smart Card Application*
The GlobalPlatform standard defines a secure smart card application (Applet) installation protocol that allows for an encrypted and hashed Applet to be installed onto a smart card. The Applet is then decrypted and verified on-card before installation. At this stage the server generates a Message Authentication Code (MAC) and encrypts (E) the CAP file (format used to distribute Applets [33]) using the shared secret between server and smart card (K_{SC})(1).

$$S: \ CAP = \text{MAC}_{K_{SC}}(Applet) \| E_{K_{SC}}(Applet) \tag{1}$$

(P2) *Preparation of Device Application*
This step of the process represents two actions; Applet injection and device application (MIDlet) security preparation. The Applet injection involves taking the byte code of the protected Applet and embedding the code into the body of the MIDlet (2). This could, for example, be done by inserting an array of byte arrays where each row represents a smart card (APDU) command. The MIDlet functionality when delivering the Applet would be to blindly send the APDUs in order. As

the Applet is encrypted and hashed the confidentiality and integrity of the item is assured.

$$S: \ JAR = MIDlet \| CAP \tag{2}$$

The MIDlet must be prepared for the secure MIDlet installation procedure defined by J2ME MIDP2.0 and implements the second security context between server and device. The server generates a RSA X.509(v3) certificate or requests one from a Certificate Authority (CA). The certificate ($Cert(MIDlet)$) is inserted into the application descriptor of the MIDlet application (3). The path of the descriptor holds all certificates necessary to validate the application except the root certificate. The Domain Protection Root Certificate ($Cert(DPRC)$) resides on the smart card and is called into play during MIDlet installation (P4). Finally, the signature of the file used to distribute MIDlets (JAR) ($S_K(JAR)$) is generated with the private key (K) of the RSA certificate, $Cert(MIDlet)$, according to the EMSA-PKCS-v1_5 encoding method of PCKS#1 version 2.0 standard. This signature is then inserted into the application descriptor (3) and the MIDlet is considered prepared.

$$S: \ JAR = JAR \| Cert(MIDlet) \| S_K(JAR) \tag{3}$$

(P3) *Server to Device Transfer*
The server can communicate to the device in a number of ways, the chosen method was to use HTTP supported by MIDP2.0.

(P4) *Device Application Installation*
This phase of the process continues to use the security context between server and device as defined in the MIDP2.0 profile. The J2ME JRE must authenticate the MIDlet application for installation into a secure domain. First the certificate ($Cert(MIDlet)$) is retrieved from the application descriptor and validated (V) against the $Cert(DPRC)$ held upon the smart card (4). The JRE then verifies the MIDlet JAR file; by taking the public key (PK) from the verified signer certificate along with a fresh SHA-1 digest ($Hash$) of the JAR file and comparing it to signature defined in the application descriptor (5). The JRE can install the MIDlet into the security domain defined by the access control model once verification is complete.

$$D: \ \mathrm{V}(Cert(MIDlet), Cert(DPRC)) \tag{4}$$

$$D: \ \mathrm{V}(S_K(JAR), Hash(JAR), PK_{Cert(MIDlet)}) \tag{5}$$

(P5) *Smart Card Installation Initiation*
The device now has a MIDlet installed in a secure domain and can communicate with the smart card. When required the Applet needs installing the device initiates communication using commands defined by the ISO 7816 specification. The device needs to authenticate itself to the card before installation can take place and the first stage is to send an APDU that issues a `GET CHALLENGE` command (6). Which returns a random challenge (r_C) from the smart card (7) [9]. The random challenge is sent to the server (8) as the response requires knowledge of the security context

between server and card (K_{SC}).

$$D \rightarrow C: \texttt{GET CHALLENGE} \tag{6}$$
$$C \rightarrow D: r_C \tag{7}$$
$$D \rightarrow S: r_C \tag{8}$$

(P6) *Server Authentication*
The server encrypts K_{SC} and r_C with K_{SC} and returns the byte string to the device (9) who sends it to the card (10) to complete a ISO7816 `EXTERNAL AUTHENTICATE (EA)` command. Once the Server Authentication as required by the GlobalPlatform has been completed the card can continue with a secure installation.

$$S \rightarrow D: E_{K_{SC}}(K_{SC} \| r_C) \tag{9}$$
$$D \rightarrow C: \mathsf{EA}(E_{K_{SC}}(K_{SC} \| r_C)) \tag{10}$$

(P7) *Checking of Smart Card Application*
The device has no knowledge of K_{SC} but does have an array of encrypted byte code representing the Applet protected by K_{SC}. The device simply sends of the APDU commands defined in this array to the smart card (11). From the smart card's perspective, a trusted card terminal has successful authenticated itself and is now, in accordance with the GlobalPlatform installation protocol, receiving an application. The on-card manager (Card Manager) receives the byte code and verifies its integrity using the Data Authentication Pattern (DAP) specified in GlobalPlatform. The verified CAP file is decrypted and installed.

$$D \rightarrow C: \text{MAC}_{K_{SC}}(Applet) \| E_{K_{SC}}(Applet) \tag{11}$$

6 Analysis

This section will examine the success of the proposed solutions in meeting the requirements defined in Section 2.3. In addition we present two scenarios that this solution could be applied to.

6.1 Requirement Analysis

All requirements defined were met by the proposed solution and industrial implementation. The first requirement was the secure deployment of device and smart card applications (R1), this has been attained. Security concerns were addressed as each element was securely transferred and installed via an individual security context. The solution is applicable to fielded devices (R2) although some configurations could be made to apply the same technology to non-fielded devices; such as pre-loading a number of device applications before devices are issued. Requirement 3 is adhered to by not affording the device the level of trust maintained between the server and smart card. The device is ignorant of the protecting secret throughout the procedure. The smart card application is encrypted at the server, decrypted on-card and it is stored as encrypted byte code whilst in the device. The confidentiality and integrity of the smart card application is

maintained (R4, R5) and further opportunities exist to take advantage of this benefit. For example, the smart card application could be used as a key distribution mechanism by the server embedding encrypted keys when preparing the smart card application for deployment.

The industrial implementation is based upon one programming language and nests existing specifications inside one another. The device application flows once to the device and once to the smart card and authenticates both entities simultaneously, when possible, by treating them as a one element. This provides a homogeneous solution (R7) and optimises resource use (R6).

6.2 Usage/Deployment Scenarios

Addressing Emerging Security Problems. Addressing an emerging security problem in the field is one usage scenario. For example, some sources [34] state that in Australia approximately 4-5% of all satellite TV subscriptions are illegal resulting in an estimated cost of $50m. The marginal increase of malicious activity [17] results in considerable financial loss for the satellite TV provider. The provider could update the security mechanisms and deploy new STB and/or smart card equipment to close the security hole but prohibitive in terms of cost and time. It would be a better option to employ the solution presented in this work to deploy a software based solution [8]. The costs involved would be lower in contrast to replacing millions of devices and the solution could be deployed at a far greater rate. The revenue streams would be secured quicker and the problem addressed. The implications of having the ability to remotely deploy complex functionality, comprising of both device and smart card applications to the field, are wide ranging.

Enhancing System Functionality. The second example exploits our solution's advantage of secure caching. A smart card is a capable entity but storage and processing power are limited. The hardware continues to improve but most cards are able to support only a limited number of third party applications and a mechanism to help manage these restricted resources would be valuable. The device can store substantially more applications depending on its storage capacity. New phones can store gigabytes compared with smart cards offering kilobytes. There is no requirement for applications that are seldom used to have its smart card application installed. When the user tries to access the uninstalled functionality, the device could install the smart card application and remove it afterwards. The server's involvement is minimal and the apparent functionality capacity of the smart card would be greatly increased. Meanwhile, the smart card application in the device cache it is protected by a higher level security context that is not available to the device or malicious attacker. Therefore, an architecture provider could roll out new functionality as the market demands and cultivate new revenue streams (m-commerce etc).

7 Conclusions

The work conducted resulted in a proposed solution that met all of the requirements defined. The method could be used for the deployment of smart card and device ap-

plications or the device element could be reduced to work purely as a deployment mechanism for smart card applications across high bandwidth channels. Advantages and deployment scenarios were presented and it is concluded that this work has posed some interesting questions for future research. For example, the concept of remotely deploying security countermeasures to fielded equipment is intriguing and could have serious application in addressing problems that may rise in the satellite TV and 3GPP mobile network in the future. This should be further explored by implementing and assessing how applicable the proposed solution is to these existing infrastructures. From an academic perspective the cryptographic processes should be fully distilled and the definition of an abstract protocol for remote application deployment outlined.

References

1. Vallée, P.: 2004 simalliance market status and 2005 outlook (2004) Presented at SIM 2005 in Amsterdam, Chairman of the Board, SIMalliance. http://www.simalliance.org/.
2. Peyret, P., Lisimaque, G., Chua, T.: Smart cards provide very high security and flexibility in subscriber management. In: IEEE Transactions on Consumer Electronics. Volume 36(3). (1990) 744–752
3. Song, W.J., Kim, W.H., Kim, B.G., Ahn, B.H., Choi, M., Kang, M.: Conditional access module systems for digital contents protection based on hybrid/fiber/coax catv networks. In: Lecture Notes in Computer Science. Volume 2869., Springer-Verlag (2003) 155–162
4. Anderson, R., Kuhn, M.: Tamper resistance - a cautionary note. In: The Second USENIX Workshop on Electronic Commerce Proceedings, Oakland, California, USENIX Association (1996) 1–11 http://citeseer.ist.psu.edu/400120.html.
5. Kömmerling, O., Kuhn, M.: Design principles for tamper-resistant smartcard processors. In: Proceedings of the USENIX Workshop on Smartcard Technology (Smartcard 99), USENIX Association (1999) 9–20 Chicago, Illinois, USA.
6. Gobioff, H., Smith, S., Tygar, J., Yee, B.: Smart cards in hostile environments. In: Second USENIX Workshop on Electronic Commerce, USENIX Association (1996) Oakland, California.
7. Cheng, S., Litva, P., Main, A.: Trusting DRM software. In: Workshop on Digital Rights Management for the Web, World Wide Web Consortium (2001) INRIA - Sophia-Antipolis, France.
8. Francis, L., Sirett, W.G., Mayes, K., Markantonakis, K.: Countermeasures for attacks on satellite tv cards using open receivers. In Buyya, R., Coddington, P., Montague, P., Naini, R., Sheppard, N., Wendelborn, A., eds.: ACSW Frontiers 2005 – Australian Information Security Workshop (AISW2005). Volume 44 of Conferences in Research and Practise in Information Technology., Australian Computer Society Inc (2005) 153–158 Newcastle, Australia. ISBN: 1-920682-26-0.
9. Rankl, W., Effing, W.: Smart Card Handbook. 3rd edn. John Wiley & Sons, Ltd (2003)
10. Aigner, M., Oswald, E.: Power analysis tutorial. Institute for Applied Information Processing and Communication, University of Technology Graz - Seminar (2000)
11. Alhussein, M.A.: Differential power analysis attack on smart card running DES. Master's thesis, Information Security Group – Royal Holloway, University of London (2004)
12. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In Wiener, M., ed.: Advances in Cryptology CRYPTO '99. Volume 1666 of Lecture Notes in Computer Science., Springer-Verlag (1999) 388–397

13. Matthews, A.: Low cost attacks on smart cards: Investigating the electromagnetic side-channel. Master's thesis, Information Security Group – Royal Holloway, University of London (2005)
14. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Investigations of power analysis attacks on smartcards. In: 1st USENIX Workshop on Smartcard Technology (Smartcard '99), USENIX (1999) 151–162 Chicago, Illinois, USA.
15. Poulsen, K.: DirectTV zaps hackers. Online Security Magazine (2001) Security Focus. http://www.securityfocus.com.
16. 3GPP TS 03.48: Technical Specification Group Terminals; Security Mechanisms for the SIM Application Toolkit; stage 2, http://www.3gpp.org. (2001)
17. Kuhn, M.: Attack on pay-TV access control systems (1997) Security seminar talk, University of Cambridge, Computer Laboratory, London, UK. http://www.cl.cam.ac.uk/ mgk25.
18. Venkatraman, J., Raghavan, V., Das, D., Talukder, A.: Trust and security realization for mobile users in gsm cellular networks. Lecture Notes in Computer Science **3285** (2004) 302–309 Springer-Verlag.
19. Badra, M., Urien, P.: Toward SSL integration in SIM smartcards. In: IEEE Wireless Communications and Networking Conference. Volume 2., IEEE (2004) 889– 893 IEEE WCNC 2004, Atlanta-Georgia USA.
20. MacDonald, J.A., Sirett, W.G., Mitchell, C.J.: Overcoming channel bandwidth constraints in secure SIM applications. In Sasaki, R., Qing, S., Okamoto, E., Yoshiura, H., eds.: 20th IFIP International Information Security Conference (SEC 2005) - Small Systems Security and Smart cards. Volume 181 of IFIP International Federation for Information Processing., Springer Science and Business Media (2005) Chiba, Japan.
21. JSR-68 JCP: Java 2 Platform, Micro Edition (J2ME) (JSR-68). Sun Microsystems, http://java.sun.com. (2002)
22. Topley, K.: J2ME In a Nutshell. In a Nutshell Series. O'Reilly (2002)
23. JSR-118 JCP: Mobile Information Device Profile, v2.0 (JSR-118). Sun Microsystems, http://java.sun.com. (2002)
24. Global Platform: Card Specification v2.1.1, http://www.globalplatform.org. (2003)
25. Global Platform: Globalplatform Device API v2.0, http://www.globalplatform.org. (2003)
26. Braentsch, M., Buhlier, P., Eirich, T., Horing, F., Oestreicher, M.: Javacard – from hype to reality. Mobile Computing - IEEE Concurrency (1999) IBM Zurich Research Labaoratory.
27. Sun Miscrosystems Inc: Runtime Environment Specification; Java Card Platform, Version 2.2.1, http://java.sun.com. (2003)
28. Sun Miscrosystems Inc: Runtime Environment Specification; Java Card Platform, Version 2.1.1, http://java.sun.com. (2003)
29. JSR-177 JCP: Security & Trust Services API (SATSA) (JSR-177). Sun Microsystems, http://java.sun.com. (2004)
30. Block, C., Wagner, A.C.: MIDP 2.0 Style Guide. The Java Series. Addison-Wesley, London (2003)
31. ISO/IEC 7816-3: Information technology - Indentifcation cards – Integrated circuit(s) cards with contacts – Part 3: Electronic signals and transmission protocols. International Organization for Standardization, http://www.iso.org. 2 edn. (1997)
32. ISO/IEC 7816-4: Information technology - Indentifcation cards – Integrated circuit cards – Part 4: Organisation, security and commands for interchange. International Organization for Standardization, http://www.iso.org. 2 edn. (2005)
33. Chen, Z.: Java Card Technology for Smart Cards: Architecture and Programmer's Guide. The Java Series. Addison-Wesley (2000)
34. Kalina, P.: No-Pay TV Costs Industry $50m. The Age Journal, http://www.theage.com.um. (2002)