A SECURE CHANNEL PROTOCOL FOR MULTI-APPLICATION SMART CARDS BASED ON PUBLIC KEY CRYPTOGRAPHY

Konstantinos Markantonakis, Keith Mayes

Information Security Group Smart Card Centre, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, United Kingdom, k.markantonakis@rhul.ac.uk, keith.mayes@rhul.ac.uk

Abstract: Smart card secure channel protocols based on public key cryptography are not widely utilised mainly due to processing overheads introduced in the underlying smart card microprocessors and the complexities introduced by the operation of a PKI infrastructure. In this paper we analyse the significance of public key secure channel protocols in multi-application smart cards. We believe that multi-application smart card technology (e.g. the GlobalPlatform smart card specification) should benefit more from the advantages of public key cryptography specifically for the initiation and maintenance of a secure channel. This paper introduces a public key based cryptographic protocol for secure entity authentication, data integrity and data confidentiality. The proposed secure channel protocol uses a combination of public key, secret key and the main idea behind the Diffie-Hellman key establishment protocols in order to achieve the desired goals.

Key words: Secure channel protocol, public key cryptography, Diffie-Hellman, GlobalPlatform, Java card, multi-application smart cards

1. INTRODUCTION

The recent introduction of multi-application smart cards has enabled cards to securely host multiple applications, dynamically and securely download or delete them at any point during the card's lifecycle. As a result, the complexity of the smart card operating system (SCOS) increased exponentially. Similarly, the complexity of the terminal applications increased significantly as new architectures [1, 2] emerged. Furthermore, as smart card technology evolves, the performance of smart card cryptographic algorithms improves and as new smart card applications are invented the benefits of public key cryptography are widely scrutinized.

Multi-application smart card technology can benefit from the use of public key cryptography both at the application level and in the SCOS level e.g. with the provision of secure channel protocols based on Public Key Infrastructures (PKI). Current versions of secure multi-application smart card standards [6] do not fully take into advantage the benefits of public key cryptography, specifically for the provision of a secure channel mechanism. The reasons range from the increased prices due to the additional processing power, up to the potentially limited performance of public key cryptographic primitives in the current generation of smart card microprocessors, or simply because there is no immediate need for such functionality.

The advantages and disadvantages of public key cryptography are widely documented in the academic literature [3, 4, 5]. In this paper we propose a public key secure channel protocol for smart cards. The protocol is based on the well known Diffie-Hellman key exchange protocol and it was designed by taking into account the processing and storage restrictions of current smart card microprocessors. Alongside with the protocol description we also provide a discussion on the operation and security requirements for its successful and efficient operation. We believe that as the number of smart card applications increases and the nature of smart card applications changes along with the differentiations on the operational requirements (e.g. dynamic application downloading and deletion), the demand for efficient smart card PKI will potentially increase.

The remainder of this paper is organised as follows. Firstly, we set up the scenery by elaborating more on the motivation behind the paper along with providing an overview of the main characteristics of a multi-application smart card standard, namely GlobalPlatform [6]. Subsequently, we highlight the main characteristics of the supporting public key infrastructure required for the successful operation of the protocol. Moving to the core idea of this paper we present the protocol details and architectural design. In order to provide a more complete coverage of the issues surrounding the implementation and operation of the proposed architecture we also provide a discussion around the security properties of the protocol by highlighting practical issues that imposed certain design decisions and directions for further research.

2. PUBLIC KEY SMART CARD SECURE CHANNEL PROTOCOLS AND THE REAL WORLD

In the following sections we provide an overview of limiting factors along with the driving forces behind the adoption of public key cryptography in multi-application smart card platforms. Similarly, we also highlight the main characteristics of a widely used multi-application smart card standard in order to provide a reference point, to the specifics of an existing architecture, along supporting the case for the existence of such a protocol.

Motivation

The advantages and disadvantages of public key cryptography have been a topic of discussion for many years. The significance of public key cryptography in smart cards, impose certain restrictions and complexities that are unique to smart card microprocessors and the nature of the infrastructures they operate.

A few years ago the main prohibiting factor for the utilization of public key cryptography in smart card microprocessors was the limited processing power of the underlying technology. However, following a number of significant improvements both at the hardware [24] and software level [20, 21, 22], the performance of public key cryptography in smart card microprocessors has improved significantly. Furthermore, the cost of a smart card microprocessor is not substantially influenced by the existence of the necessary public key functionality but rather from other factors (i.e. mainly the amount of memory).

The nature of smart card applications is also changing. Public key cryptography may be beneficial for the establishment of a secure channel when two unknown parties want to establish keys and protect subsequent communications. Such secure channels could be used for personalisation. Another use secure channels are post issuance operations, such as application/card management functions [6], protection of application or smart card operating system (SCOS) data [25].

Although the significance of public key cryptography in a smart card environment cannot be underestimated at the same time the drawbacks are not minimal. For example, a secure channel protocol designed specifically for smart cards has to be as lightweight as possible, depending of course on the underlying security and operational requirements. Furthermore, in order to improve the required performance and fulfil the security objectives a combination of cryptographic primitives and algorithms might be used. Finally, further constraints arise from the fact that often a public key based architecture requires the existence of a public key infrastructure (PKI) [26] for the management of identities, key and certificate management, etc.

Our proposed protocol aims to fulfil some of the aforementioned requirements. It is designed by keeping in mind the performance requirements and operational characteristics of smart card microprocessors. Although there is a plethora of public key cryptography secure channel protocols [3, 5, 33], most of them are not specifically designed by taking into account the specific characteristics of smart cards. For example, some cryptographic protocols although they offer more than adequate levels of security they do not keep in mind that smart card microprocessors have limited communication buffers, often ranging between 240-255 bytes. Therefore, if a protocol requires a large number of messages (e.g. key certificates) to be exchanged between the card and an off-card entity this will add to the communication and processing overheads [32]. Furthermore, the nature of a public key infrastructure requires the existence of cryptographic key certificates. For example, if a protocol requires regular checks in order to identify whether certificates are revoked or expired this might add to overall protocol security but on the other hand it will potentially complicate its mitigation in smart card environment.

The proposed solution does not claim to introduce a protocol based on new cryptographic techniques. Instead it is an implementation adaptation of existing cryptographic primitives and techniques which are carefully selected in order to be used in a smart card environment. Before moving into the details of the proposed architecture, we highlight the main characteristics of a multi-application smart card platform.

An Overview of GlobalPlatform Card Specification

In this section we highlight the main characteristics and the core components of the GlobalPlatform (GP) card specification [6], as a typical example of a multi-application smart card architecture that could benefit from the proposed protocol. Please note that among the main reasons behind the description of the GlobalPlatform architecture is that it provides the necessary functionality (e.g. secure storage of keys, key management, etc.) required by the protocol. However there are no restrictions or prerequisite for a specific type of smart card technology as the protocol could be utilised and implemented either at the application or at the (SCOS) [7, 8] level irrespectively of the characteristics of the underlying smart card microprocessor.

The GlobalPlatform smart card architecture comprises a number of on-card components that offer secure multi-application card management functionality at any given point during the card's lifecycle. Furthermore, the GlobalPlatform smart card architecture is closely coupled with the Java card [9] technology although there are no restrictions on its portability to other smart card platforms [10, 11].

The functionality provided by the underlying smart card management system includes the necessary mechanisms (e.g. secure channels [12]) that enable secure communication with the outside world. A secure channel is a mechanism that allows a card and an off-card entity to authenticate each other and establish session keys in order to protect the integrity and confidentiality of subsequent communications.

The GlobalPlatform card specification defines two protocols which are used to establish a secure channel. SCP01 is defined in Appendix D of the GlobalPlatform card specification as a symmetric key protocol that provides three levels of security (i.e. mutual authentication, integrity and data origin authentication, confidentiality). The details of the secure channel protocol (SCP02) can be found in Appendix E of the GlobalPlatform card specification. The two protocols use symmetric key cryptography for the authentication, establishment of session keys and protection of subsequent communication between the card and the outside world. While the existing protocols are mainly used for card content management purposes they can also be used by applications for secure communications. For example, secure communication between a card and an off-card entity is considered necessary whenever a sensitive operation (e.g. during cryptographic key exchanges) is about to be performed.

Another main component of GlobalPlatform is the notion of security domains. GlobalPlatform security domains are the on-card representatives of the card Issuer or an application provider. It is the security domains that allow Issuers to share control over selected portions of their card with approved partners. Additionally, security domains are responsible for cryptographic functions and key handling/separation functionality. In terms of communicating with the off-card entity in a secure way the security domains implement different secure channel protocols, as aforementioned. For the purpose of this paper, we will be using the notion of a security domain as a mechanism that will securely store keys and control access to the secure channel mechanisms.

The GlobalPlatform smart card specification is becoming the de-facto mechanism for secure application handling especially for Java cards [9] used in the GSM [28] and finance sectors [27]. There are currently ongoing discussions in order to enhance the functionality offered with the provision of additional secure channel protocols based on public key cryptography. In the following sections we present the main characteristics of the proposed protocol.

3. THE ARCHITECTURAL MODEL

In this section we highlight the main characteristics of a model for the use and operation of a public key cryptography smart card security protocol. Subsequently, we also define the main operational characteristics of the protocol.

Entities and Operation of the Model

The principal participants and relationships between participants are depicted in the following paragraphs. The main entities are off-card entities and smart cards. More specifically in a multi-application smart card usage scenario the entities that are likely to get involved in a communication session with the card are the Issuers and any Application Providers who have a business relationship with the Issuer.

For the purpose of this paper the establishment of a secure channel is divided into three sequential phases as defined in [6]:

- *Secure Channel Initiation* – when the card and the off-card entity have exchanged sufficient information enabling them to perform the required cryptographic functions. The Secure Channel Initiation phase also involved the authentication of the off-card entity by the card.
- *Secure Channel Operation* – following the exchange of card and off-card data the two entities will have the means to establish a secure channel based on recently established session keys.
- *Secure Channel Termination* – if at any stage during the operation of the secure channel either the card or the off-card entity determines that the messages received do not correspond to the expected messages or the messages do not carry the necessary cryptographic protection of expected fields then the secure channel should be terminated.

Therefore, for the purpose of this paper a secure channel is initiated either by the off-card entity using the appropriate Application Protocol Data Unit (APDU) command or by an on card entity (e.g. a Security Domain) directly when an APDU (that is cryptographically protected) is received.

Operational Characteristics

The established session keys are used for providing integrity and confidentiality on the exchanged messages. For this protocol the following requirements must be satisfied:

1. *C,* represents the smart card. Typically a sufficient tamper resistant device which is relatively difficult to compromise; it has access to a variety of cryptographic algorithms and a good random number generator. A multi-application smart card platform (e.g. GlobalPlatform) will provide significant functionality that will strengthen the overall concept of dynamic application management.
2. *H,* is a host defined as an off-card entity that requires establishing a secure channel with the smart card, application or smart card operating system (SCOS).

3. All entities share public values p and a, where p is a large prime number and a is an element of a large prime multiplicative order modulo p. We will write a^x for ($a^x \bmod p$) throughout.
4. Each card has a Diffie-Hellman key agreement key pair. More specifically, card C has private key agreement key y with corresponding public key a^y. The card's key pair can be either generated off-card by the issuer or the application provider and subsequently loaded onto the card, or it can be generated on-card (if the functionality is provided by the card). In either case the public key has to be certified by the corresponding off-card entity, i.e. the issuer or an application provider.
5. The host (H) has an RSA public encryption key, which is certified by the corresponding certification authority.
6. The card and the host share a symmetric cryptosystem and a key generation function (e.g. a one-way function) $f1(Z)$.
7. The card is capable of generating random numbers.
8. Each card (e.g. through a security domain) has a trusted copy of its owner's (e.g. certification authority, issuer or application provider) public certification key whose corresponding private key is used by the off-card entity for issuing certificates (i.e. for the Diffie-Hellman and RSA keys).

On top of these requirements the protocol should be able to fulfil the following requirements:

1. *Cheap to operate.* Its operation should not require the purchase of additional expensive smart card or host equipment.
2. *Fast.* Communication between the entities should take place with a minimal exchange of messages. Moreover the messages exchanged between the participants should minimise the use of unnecessary cryptographic operations (given the limited computational capabilities of smart cards).
3. *Efficient.* The system's operation should not restrict the normal participant's behaviour.
4. *Flexible.* It should also be able to accommodate the participant's requests for exchanging optional parameters.
5. *Secure.* It should be able to offer adequate levels of protection and follow the secure channel establishment steps as described above.

In the following section we present the architectural characteristics of the protocol.

Operational Assumptions

Given the number of the entities involved, there is clearly a need for a Public-Key Infrastructure (PKI) [29, 30] that assists these entities in managing their keys and supports the security functions of the proposed protocol.

The supporting functions of a PKI include key certification, authorisation of participating entities, and the ability of a participating entity to have multiple keys. For simplicity and in order to sustain the practicality of the overall architecture the description of the proposed infrastructure will provide examples linked with the GP architecture as described above. Furthermore, the details of the PKI infrastructure are not within the scope of this paper and we also assume that adequate key and entity management procedures are in place.

According to the proposed infrastructure, each participating off-card entity (being an Issuer or an Application Provider) has a key pair (namely certification key pair) which is used for the certification of other keys. The public key of this key pair is securely loaded on the card (e.g. in a security domain that represents the off-card entity on the card). The corresponding private key is used for the certification of RSA public encryption keys (which are used for the establishment of a secure channel). These certificates bind the included public key to the entity that is authorised to use this public key encryption key during the establishment of a secure session. As an alternative, the certification key pair might belong to a Certification Authority, which has a business relationship with the off-card entity.

Secure loading and replacement of these keys can take place by establishing a secure channel that will enable the secure transfer of keys to the card (e.g. by using the Put Key command as described in the GP specifications). Initial keys for the Issuer can be optionally hard-coded (e.g. masked in ROM) and used, during the personalisation phase, for the loading of the public certification keys. Loading of the public keys for Application Providers has to be done in a secure way (e.g. during the loading of the corresponding GP security domains or during the personalisation of these security domains). Following the loading of these certification keys, any public encryption key that belongs to an entity recognised by the security domain and is certified using the certification private key can be used for the establishment of a secure channel.

Given the proposed infrastructure, the card (or a security domain) is able to tell whether the key presented to it belongs to an entity that is authorised to establish a secure channel by verifying the certificate. For instance, if the certified key belongs to an Application Provider and is certified using the certification key loaded on the Application Provider's logical space in the card (e.g. a security domain) then the off-card entity is authorised to establish a secure session with one of the applications belonging to this Provider.

We summarise the notation used in the subsequent description of the protocol. This notation is an extended version of the notation defined in [5, 34]. Descriptions of the cryptographic algorithms appropriate for use in the protocols defined below can be found in [5, 35].

Table 1. Algorithms, Keys and Notation.

Notation	**Description**
Y||Z	Represents the concatenation of data items Y, Z in that order.
X→Y: C	Implies that entity X sends entity Y a message with contents C.
{X,Y,Z}	Implies that items within curly brackets are optional.
f1=h(Z)	IS the result of a collision resistant hash function such SHA-1 applied to the data Z.
E_K(Z)	Is the result of encipherment of data Z with a symmetric encipherment algorithm (e.g. AES or triple-DES) using key K.
PK_X(R)	Is the result of encipherment of data string R using a public key algorithm (e.g. RSA) with key X.
CSN	Represents the Card's Serial Number.
SK	Is a session key to be used for the subsequent cryptographic protection of a secure channel.
Rand_X	Is a random number generated by entity X (e.g. a Host or a Card).
Cert(X)	Represents a certificate on key X, e.g. X=Host_DH.
X_PEK	Represents entity's X Public Encryption Key, e.g. an RSA key.
X_SEK	Represents entity's X Secret Encryption Key, e.g. an RSA key.
X_DH	Represents entity's X Diffie-Hellman Public Key, e.g. Host_DH.

To strengthen the security provided by this scheme and considering that the off-card entity might use the certification key pair to certify keys not used by this protocol, certificates have to explicitly state that the certified keys are authorised to be used for the establishment of a secure channel. This explicit authorisation is granted when specified in one of the certificate extensions. Given an Issuer, who would typically have many certified keys for different purposes, there is clearly a need to protect the card from accidental or deliberate misuse of a key that is not authorised for this purpose. Therefore, the card should only use those keys that explicitly state in a dedicated extension that they can be used for communications with the card, and more specifically, for establishing a secure channel.

Apart from the off-card entities' RSA public encryption keys, the proposed protocol requires each card to have one or more Diffie-Hellman keys [12]. There are two options for the certification of these keys; either the card has a single key pair which is certified by the Issuer and shared among application (or security domains) that exist on the card, or each application (or security domain) has its own key pair certified by the entity it belongs to. The second option provides more flexibility as it allows the corresponding entity to specify the format based on their applications requirements. Given that none of these approaches introduce any risks to the security of the protocol it is up to the issuer's discretion to adopt either of these options. Please note that the infrastructure required for supporting the certification and verification of these keys or the certificate format [32] is beyond the scope of this paper.

4. A PUBLIC KEY SECURE CHANNEL SMART CARD PROTOCOL

In this section we present a technique that use well-established public key techniques for mutual authentication and key establishment between a smart card and an off-card entity based on the principles of the Diffie-Hellman key agreement protocol and a combination of symmetric and asymmetric cryptography.

The Protocol

The proposed protocol, which involves a host (off-card entity) *H* and a card *C*, consists of the following steps (please note that messages in curly brackets are considered as optional):

1. The host initiates the protocol by sending the following message to the card:

   H → C: Cert(Host_DH) || Rand_H || { Host_ID ||
   Request_Cert(Card_DH) ||
   Request_Cert(Card_PEK) ||
   Cert (Host_PEK)}

   where {*optional parameters}* is used by the host to inform the card on certain communication requirements (e.g. protecting certain card details and state whether the card has to return to the host the certificate on its Diffie-Hellman public key or just the certificate's identification number).

2. On receiving message (1) the card verifies the certificate *Cert(Host_DH)* using the preloaded public certification key of the corresponding off-card entity. If the certificate verification is successful and the entity is pre-authorised (e.g. if the entity possesses a security domain) then the card checks whether there are any optional parameters. If the are no problems with the message the card calculates *K, as* the output of a key generation function *f1* whose input is the shared Diffie-Hellman key α^{xy}, i.e. $K = f1(\alpha^{xy})$, it generates a pseudorandom number (Rand_C) and encrypts the two random numbers with key K. Subsequently, depending on the optional parameters it formulates the following message, which is optionally encrypted with the host public encryption key (Host_PEK):

   C → H E_K(Rand_H || Rand_C) { $PK_{HostPEK}$((Cert(Card_DH) || CSN) || Rand_H) }

   On receiving message (2) the host uses its private encryption key (Host_SEK) to decrypt the second part of the message. Subsequently, it verifies Cert(Card_DH) and ensures that the message comes from the required card (CSN). Subsequently, it generates key K (by using the card's Diffie-Hellman certificate) and decrypts the first part of the message in order to obtain the card's random number (Rand_C). Finally, it generates a new random value i.e. Rand_HB. The optional *session keys (SK)*, if sent to the card, will be used as the session keys for the established session. This is useful during card personalisation and card updates where the off-card system has pre-computed the messages to speed up the process. Note that if the off-card entity does not sent *session keys*, a key generating function can be utilised for the generation of session keys (which will be used to provide integrity and confidentiality for the exchanged messages). Finally, it sends the following response to the card:

   H → C E_K (Rand_C, {*SK*}, Rand_HB)

3. On receipt of the host's response the card decrypts the message and it verifies the content (i.e. the correct Rand_C); if no problems are encountered it uses the newly obtained session keys and sends the following response to the host:

C → H E_{SK}(Rand_HB, {optional parameters})

4. On receiving the message the host will use the previously established session keys in order to decrypt the message and obtain the previously sent random number (Rand_HB) along with any further optional card details.

If all the steps are successful the host and the card will use the established session keys (or the keys provided by the host in step two of the protocol) for the protection of exchanged messages throughout this session.

5. PROPERTIES AND SECURITY ANALYSIS

The proposed protocol provides mutual authentication and session key establishment between the communicating entities, i.e. an off-card entity and the card. The established session keys can be used to optionally provide integrity and message authentication as well as confidentiality on subsequent communications. Although the protocol is based on public key techniques it takes into account the restricted computing resources offered by a smart card (as briefly described in the previous sections). Therefore, the number of expensive computations (like the ones required by public key cryptography) are minimised to avoid processing overheads.

One of the factors that could affect the number of expensive computations was the choice of the Diffie-Hellman keys. Diffie-Hellman keys can be of two flavours; either long term, preferably certified, keys or just short term keys that are typically used for a single session. The card's Diffie-Hellman key pair is fixed in order to avoid the computational overhead required for the generation of a new key pair (a relatively computationally expensive operation for a smart card given that the card has this capability) for each session. However, there is nothing to prohibit a card to securely generate a new Diffie-Hellman key pair if operational security or application requirements impose this. On the other hand, it is assumed that the host possesses the computational resources for computing and storing a large number of key pairs. For that particular reason it uses a new key pair (for each communication), as opposed to a fixed certified one, so that to avoid one more certificate verification on the card. Note that the host can generate these keys in advance to avoid delays introduced by the generation of these keys during the establishment of a secure session.

What can go wrong?

Among the main issues surrounding the deployment and operation of a security protocol is the compromise of the scheme's private keys. If a card's

Diffie-Hellman key pair is compromised, it is the Issuer's decision whether to terminate or block this card, or simply update this card's Diffie-Hellman key pair. In the GP analogy if the key belongs to an Application Provider's security domain the Application Provider has to simply update this key by using the Put Key command.

If an off-card entity's (e.g. the Issuer or Certification Authority) RSA encryption key pair is compromised, the off-card entity has to perform the following actions in order to prevent further use of the compromised key by a malicious user:

1. The off-card entity has to generate a new certification key pair, which will replace the one used to certify the compromised key.
2. The off-card entity has to generate a new RSA encryption key pair and certify the public key of this key pair using the new private certification key. Note that if the issuer has issuer multiple certification keys, it then has the option not to generate a newly created key pair but to use an existing one.
3. All the cards that carry the old public certification key have to be updated with the new public key. As soon as the cards obtain the new certification key they will be able to reject certificates that were created using the compromised key.

Replacement of the certification key pair is also deemed necessary when RSA public encryption key certificates are due to expire to ensure that a key is not used beyond its expiration date. The off-card entity can use the above method to replace these keys.

An off-card entity, being the Issuer or an Application Provider, can have multiple RSA encryption key pairs to avoid unnecessary exposure of a single key. Given that the public key of this key pair is certified by a certification private key whose public counterpart is loaded on the card, the card will be able to verify this key and use it for the establishment of the secure channel. Off-card entities can also use multiple certification keys. In that case, however, the off-card entity has to have access to information that will assist it in the choice of the correct public encryption key certificate, prior to initiating the establishment of a secure channel. In the GP analogy (as defined in [6]) this information can be part of the security domain management data provided to the host as a response to the SELECT command.

Protocol Efficiency

At the very first instance it can be argued that the protocol is relatively heavy, especially when compared with corresponding symmetric key protocols. However, it is well established that the advantages that public key cryptography has to offer will have to be balanced with the anticipated processing and architectural overheads. Most of the publicly available smart card secure channel protocols are based on symmetric cryptography techniques, e.g. the GlobalPlatform ones. On the other hand a potential comparison with a number of public key secure channel protocols for devices with not some many communication and processing characteristics will not add a lot of value.

However, by taking into account the performance of cryptographic algorithms as defined in [31, 32] we can provide some indicative estimates on the performance of the cryptographic protocol, please refer to Table 2.

From Table 2 we can observe that cryptographic operations of the protocol can be completed in less than a second. Please note that this figure does not include the time spent by the SCOS to form the messages according to the protocol requirements and also to move any data from EEPROM to RAM and vice versa. Furthermore, it does not include any performance measurements for the transmission of APDUs as required by each step in the protocol. However, they give an indication as to how much time is spent in the cryptographic part of the protocol.

Table 2. Approximate Performance of the Cryptographic Protocol According to Theoretical Timings.

Operations	**Approximate Timings (ms)**
1. Two RSA signature verifications for the host certificate verification on the Host public, and Diffie-Hellman keys.	~2*160
2. A random generation (RandC).	~ 30
3. An RSA encryption for encrypting RandC, the card CSN and the key K1.	~160
4. A DH computation of a shared secret value α^{xy}	~300
5. A secret key encryption for the encryption of the card certificate cert (C-DH).	~10
6. A symmetric decryption.	~10
Totals:	**~830ms**

Furthermore, in order to successfully verify the actual performance details of the protocol we are currently, experimenting with its development in a Gemplus GemXpresso card (i.e. Java card Ver. 2.1 [18] and GP 2.1

platform [19]). We believe that in the final version of the paper we will also have obtained the required performance measurements which will be included as another section (i.e. performance measurements from a Java card implementation of the protocol) in the paper.

6. CONCLUSIONS

In this paper we have outlined the necessity and importance of using public key based cryptographic protocols for the establishment of secure channels in a multi-application smart card environment. Although public key protocols were not widely used in smart card microprocessors due to their limitations in processing power, recent technological improvements [14, 15] along with improvements in the operation of cryptographic algorithms [16, 17], make the whole idea more attractive and more feasible.

The core of this paper is dedicated in the development of secure channel establishment protocol that uses standardised public-key techniques (e.g. Diffie-Hellman) in order to provide mutual authentication and key establishment. The supporting infrastructure required to sustain the protocol's cryptographic operations is also defined. The proposed protocol, which benefits from the advantageous key management functionality provided by public key cryptography, can be utilised in a wide range of smart card microprocessors. It can be used both by the underlying SCOS and by smart card applications. More importantly, it can also be smoothly integrated in the architecture of existing multi-application smart card technologies as in the case of GP.

The future demands for public key smart card protocols will increase taking into account the needs and architectural/business models of various security sensitive applications. We are currently experimenting with the theoretical and practical implementation details around the design of public key secure channel protocols (e.g. based on elliptic curve cryptography) and also compare their performance with other existing protocols.

7. REFERENCES

1. PC/SC Workgroup. "Specifications for PC-ICC Interoperability". www.smartcardsys.com
2. OpenCard Consortium. "OpenCard Framework Specification OCF". www.opencard.org
3. B. Schneier. "Applied Cryptography", Second Edition, John Wiley & Sons, 1996.
4. W. Rankl, W. Effing. "Smart Card Handbook", John Willey and Sons, 1997.
5. A. Menezes, P. van Oorschot, S. Vanstone. "Handbook of Applied Cryptography", Boca Raton CRC Press, 1997.
6. Global Platform. "Open Platform Card Specification", Version 2.1. June 2001. http://www.globalplatform.org.

7. P.H. Hartel , E.K. de Jong Frz. “Smart Cards And Card Operating Systems”, In J. Bartlett, editor, UNIFORUM' 96, pages 725-730, San-Francisco, California, Feb 1996. Uniforum, Santa Clara, California.
8. C. Markantonakis. “The Case For A Secure Multi-Application Smart Card Operating System”, Springer-Verlag Lecture Notes in Computer Science Vol. 1396.
9. Javasoft. “Java Card Platform Specifications”, Version 2.2, September 2002. http://java.sun.com/products/javacard/specs.html
10. Microsoft. “Windows for Smart Card”. http://www.microsoft.com/HWDEV/TECH/input/smartcard/
11. MAOSCO. “MULTOS Reference Manual Ver 1.2”. http://www.multos.com/
12. International Organization for Standardization. Genève, Switzerland. ISO/IEC 7816–4, Information technology—Identification cards—Integrated circuit(s) cards with contacts—Part 4: Interindustry commands for interchange, 1995.
13. W. Diffie, M. Hellman. “New Directions in Cryptography”, IEEE Transactions on Information Theory, 22:644-654, 1976.
14. R. Ferreira, R. Malzahn, P. Marissen, J.-J. Quisquater and T. Wille. “FAME: a 3rd generation coprocessor for optimising public key cryptosystems in smart card applications”, Smart Card Research and Advanced Applications – Cardis '96, Publ. Stichting Mathematisch Centrum, pp. 59-72, 1996.
15. T. Boogaerts, "Implementation of elliptic curves cryptosystems for smart cards", CARDIS 1998, 14-16th September 1998.
16. H. Handschuh, P. Paillier. "Smart Card Cryptoprocessors for Public Key Cryptography”, In Third Smart Card Research and Advanced Application Conference – CARDIS'98, Lecture Notes in Computer Science, volume 1820, pages 372-379, Springer-Verlag, 2000.
17. L.C. Guillou, M. Ugon, J.J. Quisquater. “The Smart Card (A standardised Security Device Dedicated to Public Cryptology)”, in G.J. Simmons, Ed., Contemporary Cryptology: The Science of Information Integrity, ISBN 0879422777.
18. Gemplus. GemXpressoRAD. Gemplus, 2003.
19. Giesecke & Devrient. StarSIM Developer Suite. G&D 2003.
20. J.S. Coron, M. Joye, D. Naccache, P. Paillier. “Universal padding schemes for RSA” In M. Yung, Ed., Advances in Cryptology - CRYPTO 2002, vol. 2442 of Lecture Notes in Computer Science, pp. 226-241, Springer-Verlag, 2002.
21. J.S. Coron, D. M'Ra hi, C. Tymen. "Fast generation of pairs (k,[k]P) for Koblitz elliptic curves” In S. Vaudenay and A.M. Youssef, Eds., Selected Areas in Cryptography, vol. 2259 of Lecture Notes in Computer Science, pp. 151-164, Springer-Verlag, 2001.
22. M. Joye, P. Paillier, S. Vaudenay. “Efficient Generation of Prime Numbers" In .K. Ko and C. Paar, Eds., Cryptographic Hardware and Embedded Systems – CHES 2000, vol. 1965 of Lecture Notes in Computer Science, pp. 340-354, Springer-Verlag, 2000.
23. R. Ferreira, R. Malzahn, P. Marissen, J.J. Quisquater, T. Wille. FAME: A 3rd generation coprocessor for optimising public key cryptosystems in smart card applications, In P. H. Hartel et al., editor(s), Smart Card Research and Advanced Applications – Cardis '96, pages 59-72, Springer-Verlag, 1996.
24. UCL. “A Smarter Chip for Smart cards”. www.dice.ucl.ac.be/cascade, 1996.
25. K. Markantonakis. "Secure Log File Download Mechanisms for Smart Cards", Third Smart Card Research and Advanced Application Conference (CARDIS'98), September 14-16 1998, UCL Louvain-La-Neuve-Belgium, Lecture Notes in Computer Science, volume 1820, pages 285-304, Springer-Verlag, 2000.
26. ISO/IEC 11770-3. “Information Technology - Security Techniques - Key Management - Part 3: Mechanisms using asymmetric techniques”, ISO 1999.
27. L.G. Wang. “Smart Visa and Java Technology”, June 04, 2001. http://java.sun.com/features/2001/06/visa.html

28. 3GPP. "GSM 03.48 Digital Cellular Telecommunications System, SIM Toolkit Secure Messaging".http://www.3gpp.org/ftp/tsg_cn/WG4_protocollars/Temp/SMG%2323/TDocs/P-97-790.pdf
29. ISO/IEC 11770-1. "Information Technology - Security Techniques - Key Management - Part 1: Framework", 1996.
30. ITU-T X.509. "The Directory – Public key and Attribute Certificate Frameworks".
31. H. Handschuh, P. Paillier. "Smart Card Crypto-Coprocessors for Public-Key Cryptography", The Technical Newsletter RSA Laboratories, Vol 1, Number 1, Summer 1998.
32. K. Markantonakis. "Is the Performance of the Cryptographic Functions the Real Bottleneck?", IFIP TC11 16th International Conference on Information Security (IFIP/SEC'01), June 11-13, 2001, Paris, France, In "Trusted Information: The New Decade Challenge" , Kluwer Academic Publishers , ISBN 0-7923-7389-8, pages 77-92.
33. C. Boyd, A. Mathuria, "Protocols For Authentication and Key Establishment", Springer Verlag in Information Security and Cryptography, June 15, 2003.
34. "ISO/IEC 9798-1. Information Technology – Security Techniques – Entity Authentication – Part 1: General", 1997.
35. D.R Stinson, "Cryptography: Theory and Practice", CRC Press 1995.