IMPLEMENTATION OF A KEY EXCHANGE PROTOCOL FOR SECURE COMMUNICATION BETWEEN SIM CARD AND SERVICE PROVIDER

[1]KEREM OK, [2]CEM CEVIKBAS, [3]MOHAMMED ALSADI, [4]BUSRA OZDENIZCI, [5]VEDAT COSKUN

[1]Information Technologies Department, Isik University, Turkey
[2]Turkcell Technology, Turkey
Email: [1]vedat.coskun@isikun.edu.tr, [2]cem.cevikbas@turkcell.com.tr

Abstract: Latest Subscriber Identity Module cards are produced based on the latest specifications and can provide secure end-to-end encryption between SIM card and a Service Provider. However, un-keyed SIM cards do not contain the required security infrastructure to provide end-to-end encryption. Hence, new, emerging, or smart services those require end-to-end encryption between SIM card and a SP is impossible. Thus, providing a model to enable end-to-end encryption becomes a vital problem. In our previous work, we designed a key exchange protocol named SIMSec protocol, which creates symmetric keys by the collaborative work of un-keyed SIM card and the Service Provider's server. After a successful protocol execution, SIM card and Service Provider creates the symmetric keys and can perform end-to-end data encryption when required. In this paper, we give the details of SIMSec protocol's implementation on both SIM cards and the server.

Keywords: Key Exchange Protocol, SIM Card, Smart Card, Symmetric Encryption, Java Card.

I. INTRODUCTION

Latest SIM cards can provide secure end-to-end encryption between SIM card and Service Provider (SP), since corresponding security keys are embedded to those –keyed– SIM cards at manufacturing phase by the card issuer (CI). However, most SIM cards those dispensed to the users today do not have embedded keys that can be used for end-to-end encryption between those –un-keyed– SIM cards and SP.

End-to-end encryption between SP and the mobile SP application on SIM card is an irrevocable requirement for secure services. A SP needs to be sure that no one can modify the communication conducted with the user. Data should be appropriately encrypted using a secure protocol by a satisfactory key length. Considering the properties of the SIM cards, using symmetric encryption protocols [1-5] is favorable.

In our previous work [6], we have provided SIMSec protocol, an end-to-end key exchange algorithm between an un-keyed SIM card and an SP in which neither party is equipped with an encryption key at the beginning. When the corresponding application for SIM card is developed and OTA loaded to the SIM card with the permission of MNO, mobile SP application on the SIM card and the SP server can collaboratively create the symmetric key. As the same key is generated at both sides, SP and SIM card can perform end-to-end symmetric encryption.

Implementation of the key exchange protocol on SIM cards is traditionally troublesome, since SIM cards can only provide limited functionalities, storage, and processing power. Moreover, these SIM cards have generally Java Card 2.1 operating system which provides limited set of APIs for operations such as encryption and decryption; do not support big number operations; and they support only limited data types as short, byte, boolean and char.

In this paper, we give the details on the implementation of SIMSec protocol. The remainder of this paper is organized as follows. Section 2 includes the SIMSec protocol; we give the implementation details in Section 3; and eventually in Section 4, we conclude the study.

II. SIMSEC PROTOCOL

SIM cards are classified as keyed and un-keyed based on their possession of encryption keys immediately after their production. Keyed SIM cards are produced according to the latest GlobalPlatform Card Specifications [7], and personalized during manufacturing phase to include required private keys. This situation enables sensitive services -such as digital signatures and mobile financial services- after the SIM card is issued to a user. On the contrary, un-keyed SIM cards do not have any previously installed private keys to be used by the off-card entities for providing secure communication or any related services. Some issues on this problem are also given in [6, 8]. Handling of keyed and un-keyed SIM cards are managed to the customers based on the intended services; keyed cards with integrated key and higher capability –and with higher cost as well- are used for secure services, but un-keyed cards –with lower capability and lower cost- are sufficiently capable for regular services. The problem arises when a user requests unsecure services only at the beginning -and receives an un-keyed SIM card accordingly- but requests a secure service after a while. This situation, as a matter of fact, constitutes our research problem area.

In keyed SIM card situation, (See **Fig.1**), issuance of the SIM cards to the users by the MNOs is the first

step. As an SP wishes to offer a secure service via a –keyed– SIM card, it makes an agreement with the MNO to use a specific slot –say, slot n– of the SIM card. After the agreement is signed, MNO notifies CI about the agreement. Then, CI shares the corresponding slot key with the SP. After SP gets the key, it can install applications to the slot *n* of the SIM card using the key; and SP can exchange data with the application running in the specified slot securely by using the slot key (K_n). Please note that the described communication protocol steps are flexible, hence may be modified based on the standards and specifications that a specific card follows, since each card may be produced according to a specific specification.

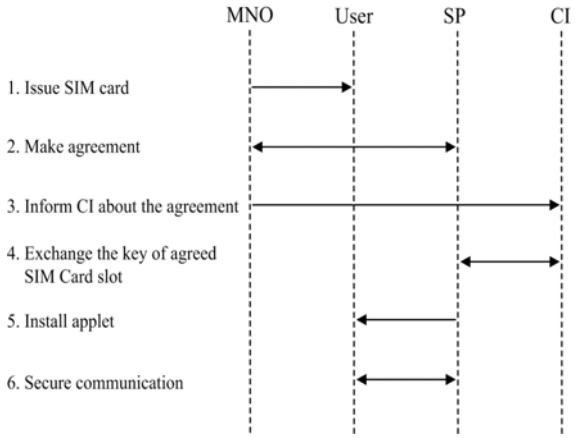

Fig.1. Secure Communication between SP and SIM card in keyed SIM cards

In our previous study [6], we designed SIMSec key exchange protocol, which provides end-to-end encryption between the SP and the SIM card. After successful implementation of this protocol, SPs will be able to offer value added secure services to the users. For this purpose, the SIMSec SIM card application and SIMSec server application that implements the SIMSec protocol establish the secure data exchange infrastructure without an additional cost or effort.

In our service usage conceptual model (See **Fig.2**); a user firstly requests a service that a SP offers; after which the SP informs the MNO about the request. The MNO OTA installs the SIMSec Card application to the SIM card, which will enable generation of symmetric key between SIM card and SP. In the last step, the SIM card and the SP generate the keys interactively using the SIMSec Card application on the SIM card and the SIMSec Server application on the SP's server. The developed protocol is used in Step 4 of the conceptual model. After the exchange of the keys, both sides will be able to provide end to end encryption.

III. IMPLEMENTATION OF SIMSEC PROTOCOL

Implementation of SIMSec protocol is performed on both SIM card and the server. On SIM cards, the implementation of the protocol is performed using Java card 2.1 programming language environment that exists in un-keyed SIM cards. Using predefined high-level functions increases efficiency of the potential application, but primitive operations are not efficient at all. The protocol is designed for the SIM cards to make use of already existing crypto primitives such as DES, random data generator, and message digest; so that not many low level operations are required. Even so, we had to use such low level operations in calculating the Diffie-Hellman values because of lack of such function in crypto primitives.

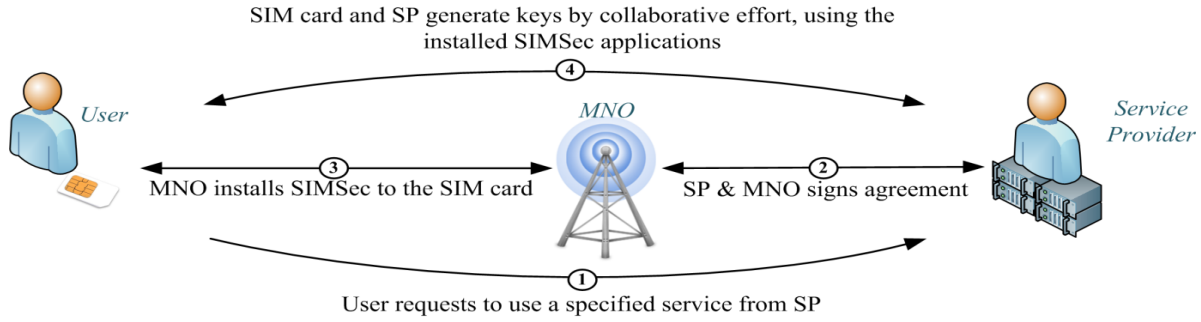

Fig.2. Conceptual model of the SIMSec protocol

The most difficult part of the protocol implementation was calculating the Diffie-Hellman values. As we aimed to calculate g^a (mod p) and $(g^b)^a$ (mod p) values by using low level mathematical primitive functions, we ended up with very high time requirements. Please consider that that the length of the *g* value is 8 bits; the length of *a* value is 384 bits; and the length of the *p* value is 1024 bits. Therefore, the only solution was to use already existing crypto primitives for these calculations [6].

When RSA encryption is investigated in detail, it was seen that performing an RSA encryption is very similar to calculating an exponentiation in Diffie-Hellman. In RSA encryption, when user B wants to encrypt a message m, she needs to calculate m^e (mod n) in which the n and e values are public keys of user

A. Thus, Diffie-Hellman exponentiation can be performed using RSA encryption crypto primitive. In order to calculate the Diffie-Hellman values on Java card, following mappings are used:

- For the message content (m value) in RSA encryption, we used g value of Diffie-Hellman,
- For the first public key of user A (e value) in RSA encryption, we used g value of Diffie-Hellman,
- For the second public key of user A (n value) in RSA encryption, we used p value of Diffie-Hellman.

In the protocol, we used the RSA encryption functions to calculate the Diffie-Hellman values using the crypto primitives provided in Java card. The details of matching Diffie-Hellman values in RSA encryption is given in **Table 1**. Related code part that is used for calculating Diffie-Hellman values using RSA encryption is also given in **Fig.3**.

Table1: Matching Diffie-Hellman values in RSA Encryption

	Base Value	**Exponent**	**Mod**	**Result**
RSA Encryption	m	e	n	$m^e \pmod n$
1st Diffie-Hellman	g	a	p	$g^a \pmod p$
2nd Diffie-Hellman	$g^b \pmod p$	a	p	$(g^b)^a \pmod p$

```
Cipher cipher = Cipher.getInstance(Cipher.ALG_RSA_NOPAD,false);
RSAPublicKey publicKey = (RSAPublicKey)
    KeyBuilder.buildKey(KeyBuilder.TYPE_RSA_PUBLIC,
    KeyBuilder.LENGTH_RSA_1024, false);
publicKey.setExponent(a, (short) 0,(short) a.length);
cipher.init(publicKey, Cipher.MODE_ENCRYPT);
cipher.doFinal(g, (short) 0, (short) g.length, ga, (short) 0);
```

Fig.3. Java card programming code for calculating Diffie-Hellman values

The rest of SIMSec protocol is implemented using the crypto primitives those reside on SIM cards. MessageDigest class is used to perform hashing functions; RandomData class is used to generate random numbers securely; SIMView class is used to read IMSI and ICCID numbers from SIM cards. The communication between SIM card and server is performed using the SMS channel. Thus, corresponding SMS classes are also implemented in the protocol.

CONCLUSIONS

In our previous work [6], we presented SIMSec symmetric key exchange protocol between an un-keyed SIM card and SP in which both parties are not equipped with an encryption key at the beginning. In this paper we presented the implementation of SIMSec protocol. When the implemented applications are executed, a symmetric key is collaboratively generated by the SIM card and the SP. Then, they can use the generated key for end-to-end encryption. The implementation of the protocol consequently enables SIM card and SP to establish and facilitate secure and emerging applications via SIM cards such as NFC payment.

ACKNOWLEDGMENTS

This work is funded by TÜBİTAK (The Scientific and Technological Research Council of Turkey, www.tubitak.gov.tr/en) and Turkcell Technology (http://www.turkcellteknoloji.com.tr/) under TÜBİTAK project grant number 1505 - 5130053.

REFERENCES

[1] B. Schneier, “Description of a new variable-length key”, 64-bit block cipher (Blowfish). In Fast Software Encryption, 1994, pp. 191-204.

[2] W. Stallings, W, “The advanced encryption standard,” Cryptologia, vol. 26(3), pp. 165-188, July 2002

[3] D. Coppersmith, “The Data Encryption Standard (DES) and its strength against attacks,” IBM journal of research and development, vol 38(3), pp. 243-250, May 1994

[4] W. C. Barker, E. Barker, “NIST Special Publication 800-67 Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher Revision 1”, 2012.

[5] V. Coskun, K. Ok, B. Ozdenizci, “Near Field Communication (NFC): From Theory to Practice”, John Wiley & Sons., 2011.

[6] K. Ok, V. Coskun, C. Cevikbas, B. Ozdenizci, Design of a Key Exchange Protocol between SIM Card and Service Provider, 23rd Telecommunications Forum TELFOR 2015, 24-26 November 2015, Belgrad, Serbia.

[7] GlobalPlatform. Available: http://www.globalplatform.org/ (Accessed on 15 October 2015).

[8] K. Ok, V. Coskun, R. C. Cevikbas, “Challenges and Risks for a Secure Communication between a Smartcard and a SP through Cellular Network,” International Journal of Advances in Computer Networks and Its Security, vol 4(4), pp. 26-30, December 2014.

★★★