Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 70-77, Kasım 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 70-77, November 2020

Research Article

Development of a SIM Card based Key Management System

Büşra Özdenizci Köse[1*], Cem Çevikbaş[2], Hacı Ali Mantar[3], Vedat Coşkun[4]

[1] Gebze Technical University, Faculty of Business Administration, Department of Management, Kocaeli, Turkey (ORCID: 0000-0002-8414-5252)
[2] Turkcell Technology, İstanbul, Turkey
[3] Gebze Technical University, Faculty of Engineering, Department of Computer Engineering, Kocaeli, Turkey (ORCID: 0000-0002-1066-9942)
[4] Beykent University, Faculty of Engineering-Architecture, Department of Computer Engineering, İstanbul, Turkey (ORCID: 0000-0003-3052-9821)

(International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2020 – 22-24 October 2020)

(**DOI:** 10.31590/ejosat.818711)

ATIF/REFERENCE: Özdenizci Köse, B., Çevikbaş, C., Mantar, H. A. & Coşkun, V. (2020). Development of a SIM Card based Key Management System. *European Journal of Science and Technology*, (Special Issue), 70-77.

Abstract

Today, almost all applications running on the smartphone provide valuable and sensitive transactions on user's private data such as identity data, credit card details, payment data, location data and so on. Service providers are trying to increase the efficiency of their applications and to improve the compatibility of security mechanisms through SIM (Subscriber Identity Module) cards owned by Mobile Network Operators (MNOs). In this sense, there is an urgent need for a centralized secure key management service. Thanks to the strong security infrastructure created by card manufacturers during design and the robust security procedures applied by MNOs; SIM cards have significant potential to provide the required secure environment for service providers. Accordingly, a novel centralized SIM card based key management framework called SIM-GAYS is designed and developed to facilitate centralized cryptographic operations of diverse mobile applications provided by service providers. This paper aims to present and demonstrate the essential development results of SIM-GAYS. The functional requirements of the SIM-GAYS are tested depending on developed scenarios. The results showed that the designed and developed APDU (Application Protocol Data Unit) transmission method between SIM card and smartphone, and secure storage of the keys by the Master Key (owned by SIM-GAYS) support almost all cryptographic services provided by SIM-GAYS framework.

Keywords: Smartphone, Mobile, Smart Card, SIM Card, Key Management System.

SIM Kart Tabanlı Anahtar Yönetim Sisteminin Geliştirilmesi

Öz

Günümüzde akıllı telefon üzerinde çalışan hemen hemen tüm uygulamalar, kimlik verileri, kredi kartı bilgileri, ödeme verileri, konum verileri gibi kullanıcının özel bilgileri üzerinde değerli ve hassas işlemler sağlamaktadır. Servis sağlayıcılar, Mobil Ağ Operatörlerinin (Mobile Network Operator, MNO) sahip olduğu SIM (Subscriber Identity Module) kartları aracılığıyla, uygulamalarının verimliliğini artırmaya ve güvenlik mekanizmalarının uyumluluğunu iyileştirmeye çalışmaktadır. Bu kapsamda, merkezi bir güvenli anahtar yönetimi hizmetine ihtiyaç bulunmaktadır. Kart üreticilerinin tasarım sırasında oluşturduğu güçlü güvenlik altyapısı ve MNO'ların uyguladığı sağlam güvenlik prosedürleri sayesinde; SIM kartlar, servis sağlayıcılar için gerekli güvenli ortamı sağlamak için önemli bir potansiyele sahiptir. Bu doğrultuda, servis sağlayıcılar tarafından sağlanan çeşitli mobil uygulamaların merkezileştirilmiş kriptografik işlemlerini kolaylaştırmak için SIM-GAYS adı verilen yeni bir merkezi SIM kart tabanlı anahtar yönetimi çerçevesi tasarlanmış ve geliştirilmiştir. Bu çalışma, SIM-GAYS'ın temel geliştirme sonuçlarını sunmayı ve göstermeyi amaçlamaktadır. SIM-GAYS'ın işlevsel gereksinimleri, geliştirilen senaryolara göre test edilmiştir. Sonuçlar, SIM kart ile akıllı telefon arasında tasarlanan ve geliştirilen APDU (Uygulama Protokolü Veri Birimi) aktarım yönteminin ve SIM-GAYS'ın sahip olduğu Ana Anahtar ile diğer anahtarların güvenli depolanmasının, SIM-GAYS çerçevesi tarafından sağlanan kriptografik hizmetleri desteklediğini göstermektedir.

Anahtar Kelimeler: Akıllı Telefon, Mobil, Akıllı Kart, SIM Kart, Anahtar Yönetim Sistemi.

[*] Corresponding Author: Gebze Technical University, Faculty of Business Administration, Department of Management, Kocaeli, Turkey, ORCID: 0000-0002-8414-5252, busraozdenizci@gtu.edu.tr

1. Introduction

In recent years, with the development of smartphones and smart cards, the content and benefits of mobile applications offered by service providers have also been enriched. Almost all applications running on the smartphone provide valuable and sensitive transactions, in other words, contain exchange of the user's private data such as financial data, identity data, credit card details, payment data, location data, social media data and so on. It is obvious that, such sensitive data of users on smartphone must be safeguarded and protected against unwarranted disclosure. Services providers should take necessary countermeasures in order to secure the data exchange between the service application on the smartphone and their own server (Ozdenizci et al., 2019; Ozdenizci et al., 2020). In order to perform a secure communication design and minimize the risks between the smartphone user and service provider; different security mechanisms such as key management models, OTP (One-Time Password) services and encryption methods are implemented for mobile services. The encryption keys and related algorithms must be kept securely on mobile applications installed by the service providers on the smartphone (Ok et al., 2015; Ok et al., 2016).

Thanks to the strong security infrastructure created by the card manufacturer during design and the security procedures applied by Mobile Network Operators (MNOs); SIM (Subscriber Identity Module) card, the smart card of smartphone, have a significant potential to provide a secure environment. Without MNO approval, smartphone user is not allowed to install an application or to save any data onto the SIM card. When the capacity of the SIM card is available and MNO permission is obtained, additional value-added mobile applications can be installed onto the SIM card (Borst et al., 2001; Alliance, 2016; Ok et al., 2016). In this sense, the security needs of mobile services make SIM cards an important candidate for providing desired security infrastructure.

It is evident that as processing and storage capabilities of SIM cards increase, the complexity of managing security keys on the SIM cards increase. In this sense, service providers need to increase the efficiency of their applications and improve the compatibility of security mechanisms through SIM cards owned by MNOs. Recently there is definite need for a centralized secure key management service and OTP service on SIM card which was introduced briefly in previous studies (Ozdenizci et al., 2019; Ozdenizci et al., 2020).

For this purpose, a novel framework called SIM-GAYS in other words, a centralized SIM based key management framework is proposed to centralize and facilitate cryptographic operations (such as asymmetric key generation, symmetric encryption, verification of signed data and many other) of diverse mobile applications provided by various service providers. In addition, the centralized key management system aims to provide OTP generation and validation capability on SIM cards. The proposed SIM-GAYS framework is also supported by Turkcell Technology A.S. and TUBITAK (The Scientific and Technological Research Council of Turkey) under 1505 Program. This comprehensive key management system on SIM cards supports new value added services development as well as of mobile ecosystem advancement. In the previous studies (Ozdenizci et al., 2019; Ozdenizci et al., 2020), the system analysis and design considerations of SIM-GAYS framework have been presented clearly.

This paper aims to demonstrate and highlight the development issues of SIM based key management framework. The functional requirements of the SIM-GAYS system are realized and tested depending on developed scenarios. Accordingly, the rest of this paper is organized as follows: Section II reviews the system development method including system components of SIM-GAYS. Afterwards, Section III presents the development results of system functional requirements of SIM-GAYS. Finally, the study is concluded and further work is emphasized in Section IV.

2. Material and Method

The key management system on SIM card communicates with smartphone applications and SIM applets in order to fulfill encryption, decryption and other tasks of smartphone and SIM card applications. Currently the number and type of keys used on the SIM cards vary according to the security requirements of mobile applications. The developed framework aims to work with all existing smart card types, communicates with each key applet in the requested format and fulfills different communication requirements of each key applet. When a SIM Applet or a Smartphone application -such as a mobile ticketing application- needs to encrypt or decrypt a data –such as PIN or credit card number- or wishes to access a service provided by a key applet, it will access the SIM-GAYS module embedded on SIM card. Another important module of SIM-GAYS is OTP generator that serve for all service providers to facilitate the OTP operation of users and service providers. The SIM-GAYS framework and its general structure is illustrated in Figure 1. The model includes the following components and actors in order to realize the centralized key management operations on SIM cards: User, MNO, Service Provider, SIM-GAYS, SIM-OTP, OTP Validator, Service Provider Application and SIM Applet.

(1) User is the actor who uses the SIM card and utilizes the mobile service of the service provider over the SIM.
(2) MNO owns and manages the SIM card, and is responsible for OTA (Over-the-Air) communication on SIM card.
(3) Service Provider provides mobile services to the user via smartphone.
(4) SIM-GAYS is the framework application which is developed on the SIM card, and performs secure key management operations on the SIM card.
(5) SIM-OTP is the application that provides OTP generation service on SIM card. OTP Validator is the server to verify OTPs produced on the SIM card; which is deployed on a server by MNO.

(6) Service Provider Application is the mobile service (such as mobile banking, social media, e-government etc.) of the service provider on the smartphone. This application communicates with the SIM card to create keys, perform encryption or decryption operations.
(7) SIM Applet are other applications available on the SIM card. They can also make key management requests to the SIM-GAYS application.

For developing the system functions of SIM-GAYS framework; the most appropriate tools are selected. Java Card Development Kit is used which includes the required libraries for developing software in the Java Card language. In terms of Integrated Development Environment (IDE), Izy NFC and Netbeans tools are used for Java Card software development. In terms of SIM Card software installation tool, JLoad application is used to install the developed Java Card programs to the SIM card. Finally, SIM card readers (Gemplus Smart Card Readers) is used to load the developed applets and GemXpresso JCard Manager software by Gemplus is used to test SIM-GAYS in the computer environment. In addition, the developed SIM-GAYS software and OTP algorithm on the SIM card to be developed comply with the international GSM standards: GSM 11.14-SIM Application Toolkit (SAT) for the Subscriber Identity Module (SIM-ME) (3GPP, 1999a) and GSM 3.48-SIM Toolkit Security (3GPP, 1999b).

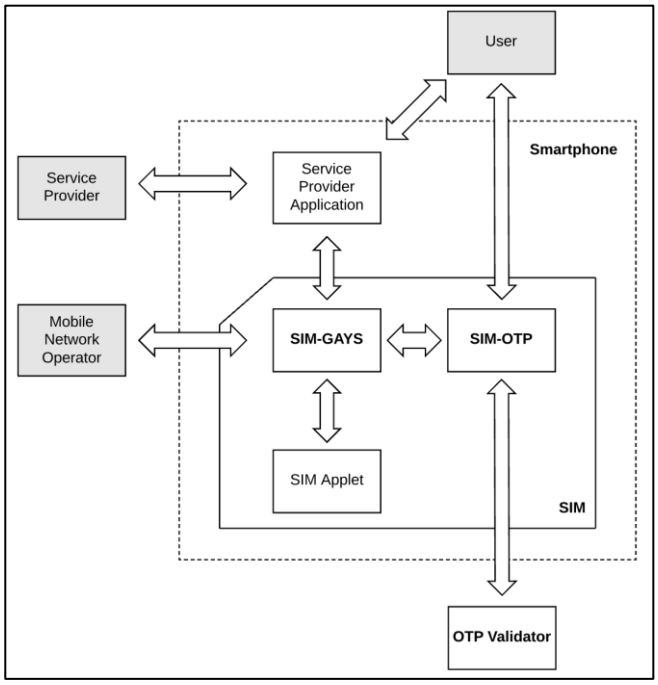


Fig. 1. Context Diagram of SIM-GAYS

3. Results

In order to realize SIM-GAYS framework on SIM cards, first the multiple APDU (Application Protocol Data Unit) message transmission between SIM card and smartphone is developed and tested successfully. Afterwards, the generated keys (i.e., asymmetric private key and symmetric key) are encrypted with Master Key of SIM-GAYS, and tested successfully.

3.1. Multiple APDU Transmission on SIM-GAYS

Messages from mobile device to SIM card (i.e., SIM-GAYS module) communications is generally too large to fit in a single APDU message. In this case, more than one message must be sent from the mobile device for a specific operation. Even if these messages are sent one after the other to the SIM card, the previous messages must be stored on the SIM-GAYS in this process.

For this purpose, a byte array totalParams shown in the code fragment (Figure 2) is created. The incoming messages are stored in totalParams array and processed later. At the same time, in the APDU message, P1 and P2 values are used as values that indicates the key length and the data length. However, since the number sizes that can be used in P1 and P2 data are not sufficient, the result obtained with the quartile of the data is sent from the mobile device in an APDU message. As a result, the SIM card reaches the accurate length by converting the received byte data into short data and then multiplying it by four.

SIM-GAYS (i.e., SIM card) needs to also transmit multiple APDU messages to the mobile device when necessary. Therefore, after the SIM card reaches a result for a specific operation, it must send the result of corresponding operation piece by piece to the mobile device, in accordance with the requests performed by the mobile device. The code fragment developed is shown in Figure 3.

3.2. Asymmetric Key Operations

Asymmetric key operations of SIM-GAYS framework include asymmetric key generation, encryption and decryption. A service provider application can request one of these operations in accordance with their specific operations. After generation of asymmetric key, the key is divided into modules and exponent values; by this way the asymmetric key can be transmitted to the mobile device as

bytes. The private key -in the fragmented key data- is encrypted together with the PIN and transmitted to the mobile device. The public key is transmitted without encryption.

```
// Receive data from mobile device

byte[] buffer = apdu.getBuffer();
wrappedKeyLength = (short) (buffer[ISO7816.OFFSET_P1] * 4);
dataLength = (short) (buffer[ISO7816.OFFSET_P2] * 4);
len = apdu.setIncomingAndReceive();
for (i = 0; i < (short) (len); i++) {
    temp256_1_[i] = buffer[(short) (i + 5)];
}

// Receive current message queue
// Receive total message size
current_m = temp256_1_[0];
total_m = temp256_1_[1];

// Perform messages and save to totalParams
if(current_m <= total_m && current_m<= (short) 3){

    if((short) (len) >= 3)
        Util.arrayCopy(temp256_1_, (short) 2, totalParams,(short) (250*(current_m-1)),(short) (len)-2);
    } else if (current_m==(short) 3){
        if((short) (len) >= 3)
            Util.arrayCopy(temp256_1_, (short) 2,totalParams, (short)(250*2),(short)(len)-2);
    }

if (current_m == total_m) {

    // Perform operations

}

// When operation ends, send results to mobile device
} else if (current_m > total_m) {

    if(Util.arrayCompare(FourBytePin1, (short) 0, FourBytePin2,(short) 0, (short) 4)==0)
      sendResultApdu(apdu, totalParams,(short)(current_m - total_m + 1),(short) (dataLength-privateKeyLength-24));

    else
      sendResultApdu(apdu, RSA_GENERATION_EXCEPTION,(short) 1, (short) (3));
}
```

Fig. 2. Multiple APDU Transmission Management by Mobile Device

```
//Invoke method
sendResultApdu(apdu, totalParams,
(short) (current_m - total_m + 1),
(short) (dataLength - privateKeyLength - 24));

public void sendResultApdu(APDU apdu, byte[] data, short seq, short dataLen)
{
    byte[] buffer = apdu.getBuffer();
    if (dataLen == 0) {
        total_length = (short) data.length;
        compare = (short) data.length;
    } else {
        total_length = (short) dataLen;
        compare = (short) dataLen;
    }

    //short block_size = (short) 253;
    divide = (short) (total_length / (short) 253);
    if (divide == 0)
        iterations = (short) (1);
    else
        iterations = (short) (divide + 1);

    if ((short) (seq * 253) > (short) (compare)) {
        ...
    } else {
        ...
    }
}
```

Fig. 3. Multiple APDU Transmission Management by SIM Card

After generating an asymmetric key on the SIM card, the key needs to be requested from the SIM card. Due to the memory limitations of SIM card, the asymmetric key cannot be stored continuously. The generated asymmetric key can lost if a new asymmetric key is generated. Therefore, the produced asymmetric key must be requested and taken from the SIM card. The arrival of the encrypted key data to the SIM card as a result of the request from the SIM card is tested and demonstrated in Figure 4.

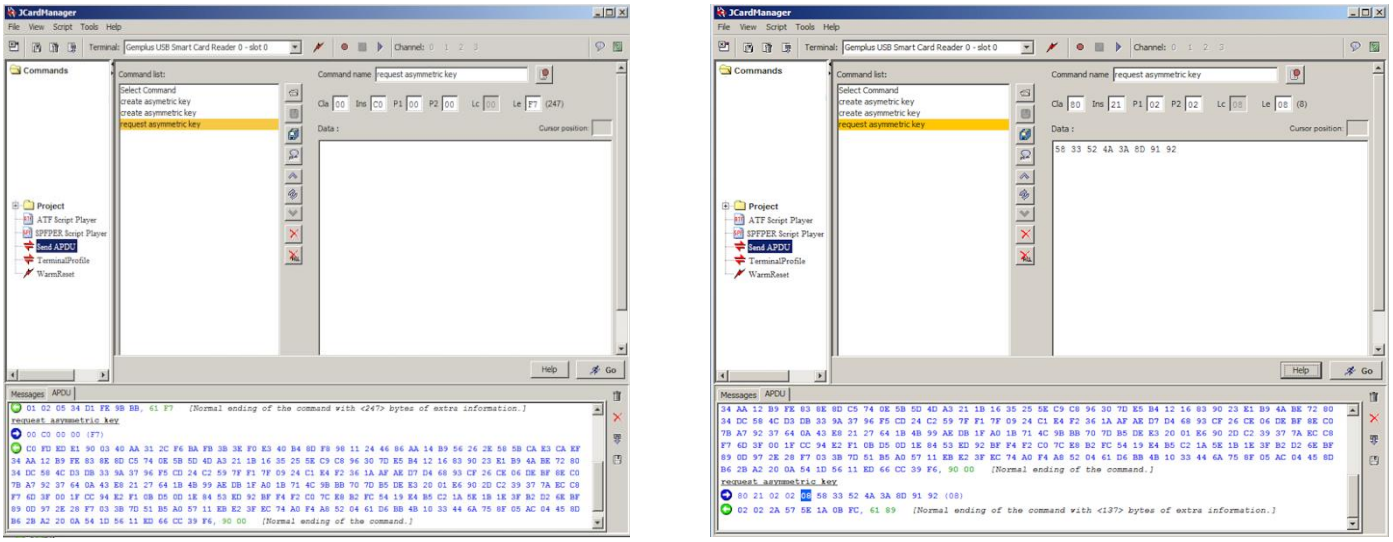
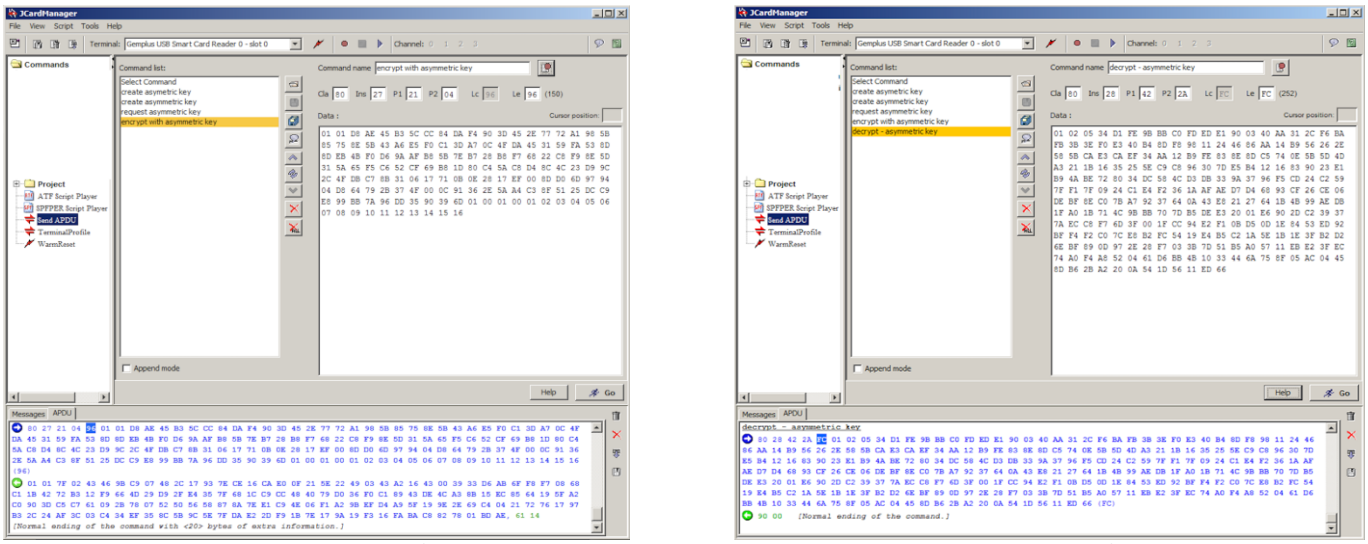


Fig. 4. Request Asymmetric Key Operations by GemXpresso JCard Manager

In case of asymmetric data encryption, first the asymmetric key is created using the mode and exponent values sent from the mobile device, and then used in the encryption process. The encryption process is carried out as a hybrid process. In other words, a 3DES (Triple Data Encryption Algorithm) key (24 bytes) is created; and the 3DES key is encrypted with the asymmetric key; and the data is encrypted with the 3DES key. In case of asymmetric data decryption, first the encrypted asymmetric secret key sent from the mobile device is decrypted; PIN check is performed respectively; and it is re-created using the mode and exponent values. Finally, encrypted data is decrypted using the asymmetric key. Figure 5 (a) and Figure 5 (b) show scenarios on asymmetric encryption of data and asymmetric decryption of encrypted data.



(a) Asymmetric Encryption of Data

(b) Asymmetric Decryption of Encrypted Data

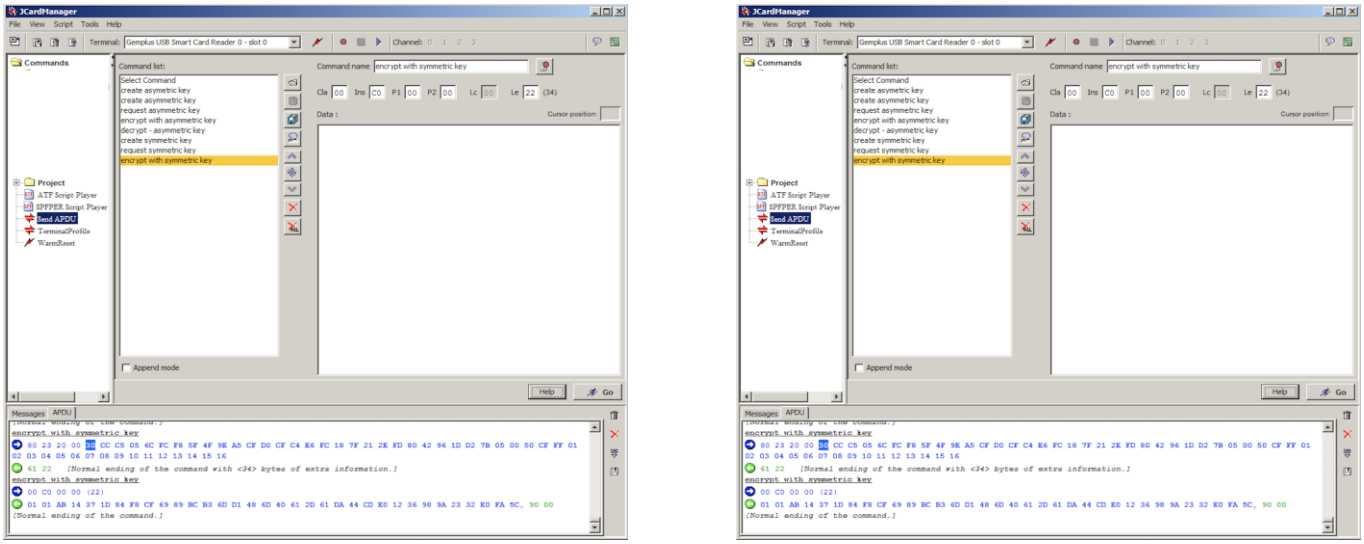
Fig. 5. Asymmmetric Key Operations by GemXpresso JCard Manager

3.3. Symmetric Key Operations

Symmetric key operations of SIM-GAYS framework include symmetric key generation, encryption and decryption. A service provider application can request one of these operations in accordance with their specific operations. SIM-GAYS application performs encryption and decryption operations by using 3DES method. An example code sent to the SIM card related to symmetric encryption and the response given by the SIM card are shown in Figure 6 (a). The encrypted symmetric key and 16 bytes of data between 01-16 were transmitted to the SIM card for encryption. The SIM card firstly decrypted the encrypted symmetric key and reached the original symmetric key and then encrypted 16 bytes of data. Also, an example code sent to the SIM card related to symmetric decryption and the response given by the SIM card are shown in Figure 6 (b).

3.4. Key Encryption and Decryption for Storage

Generated asymmetric private keys and symmetric keys, are encrypted with a Master Key generated by SIM-GAYS will be used for encryption of other keys. In this way, the mobile device is able to securely store the key sent without reducing the security level. When an encryption or decryption process with the relevant key is required, the mobile device sends the encrypted key to SIM-GAYS. Thus, SIM-GAYS will decrypt the key with its Master Key and perform the requested encryption or decryption process. The code fragment of key encryption and decryption is given in Figure 7.



(a) Symmetric Encryption of Data (b) Symmetric Decryption of Data

Fig. 6. Symmmetric Key Operations by GemXpresso JCard Manager

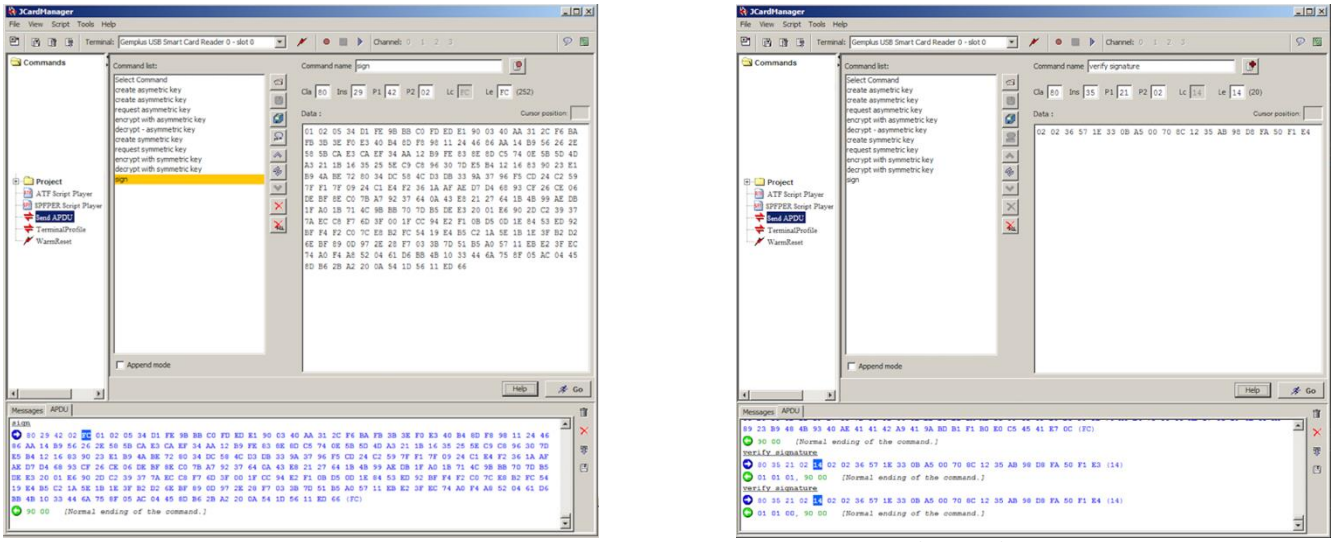
```
//Create Master Key
private byte[] masterKey = new byte[24];
DESKey deskey = (DESKey)
KeyBuilder.buildKey(KeyBuilder.TYPE_DES,KeyBuilder.LENGTH_DES, false);
rd = RandomData.getInstance(RandomData.ALG_PSEUDO_RANDOM);
rd.generateData(masterKey, (short) 0, (short) 24);

public short wrap_new(byte[] data, short length){
// Use DES key to encrypt Master Key
   deskey.setKey(masterKey, (short) 0);
    cipherCBC.init(deskey, Cipher.MODE_ENCRYPT);
// Perform encryption operations of data sent to method
    temp2=cipherCBC.doFinal(data, (short) 0, (short) length, temp_768_1_, (short) 0);
    deskey.setKey(masterKey, (short) 8);
    cipherCBC.init(deskey, Cipher.MODE_DECRYPT);
    temp2=cipherCBC.doFinal(temp_768_1_, (short) 0,(short) temp2, temp_768_1_, (short) 0);
    deskey.setKey(masterKey, (short) 16);
    cipherCBC.init(deskey, Cipher.MODE_ENCRYPT);
    temp2=cipherCBC.doFinal(temp_768_1_, (short) 0,(short) temp2, temp_768_1_, (short) 0);
    return temp2;
}

// Decrypt the keys
public short unwrap_new(byte[] data, short len) {
    deskey.setKey(masterKey, (short) 16);
    cipherCBC.init(deskey, Cipher.MODE_DECRYPT);
    temp2 = cipherCBC.doFinal(data, (short) 0, (short) len, temp_768_1_, (short) 0);
    deskey.setKey(masterKey, (short) 8);
    cipherCBC.init(deskey, Cipher.MODE_ENCRYPT);
    temp2 = cipherCBC.doFinal(data, (short) 0, (short) temp2, temp_768_1_, (short) 0);
    deskey.setKey(masterKey, (short) 0);
    cipherCBC.init(deskey, Cipher.MODE_DECRYPT);
    temp2 = cipherCBC.doFinal(data, (short) 0, (short) temp2, temp_768_1_, (short) 0);
    return temp2;
}
```

Fig. 7. Key Encryption and Decryption on SIM-GAYS

3.5. Signing Operations

Service provider application can request signing a data from SIM-GAYS application using the asymmetric private key. In addition, service provider application can request the verification of the data -previously signed by SIM-GAYS- from SIM-GAYS application. In this case, public key is created from the mode and exponent value; and then signature is verified by public key. The

encrypted asymmetric private key sent from the mobile device is first decrypted and then PIN value is checked; and re-created using the mode and exponent values. Finally, the data sent is signed with the private key. For the verification of signed data; similarly, the asymmetric key is first created using the mode and exponent values sent from the mobile device, and then the signature verification process is performed. The example data sent to the SIM-GAYS for signing and verification operations are presented in Figure 8.

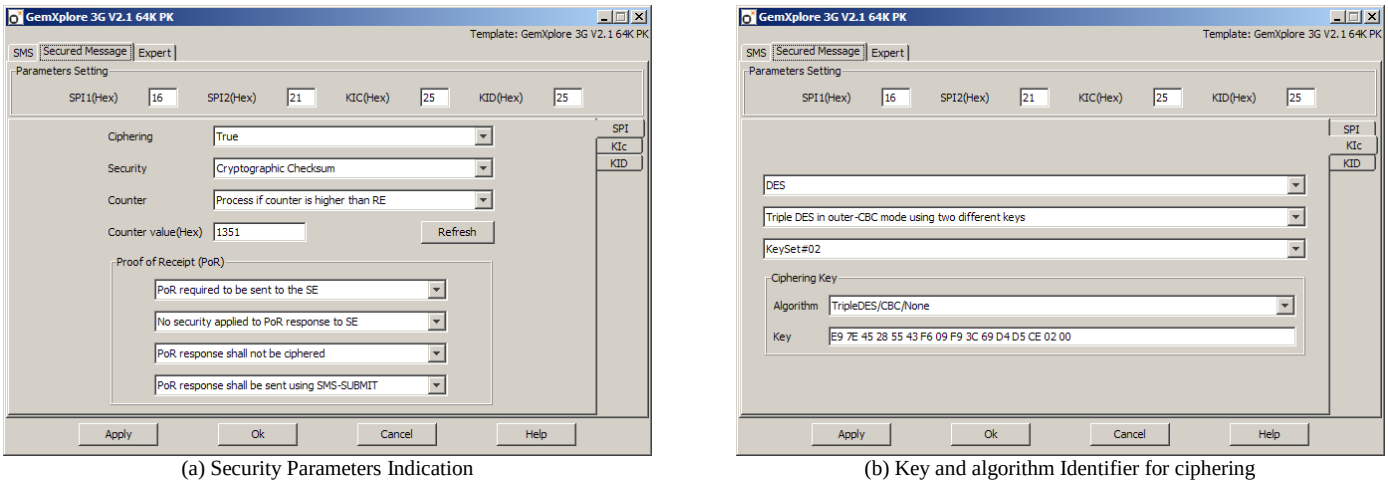


(a) Signing on SIM-GAYS (b) Verification of Signature on SIM-GAYS

Fig. 8. Signing operations by GemXpresso JCard Manager

3.6. OTP Operations

SIM-GAYS system also provides SIM-OTP services. Service provider application can request OTP generation using the SIM-OTP application. Accordingly, Service Provider requests OTP from user; and the user transmits the request to generate OTP to SIM-OTP. SIM-OTP application receives the request of the user; and then produces OTP for the user. The OTP data is also verified with the help of SIM-OTP application. SIM-OTP sends OTP to the Service Provider's server by encrypting it with 3DES (with a shared symmetric key with the Service Provider). Service Provider receives the encrypted data and accesses OTP by decrypting the encrypted data with 3DES. The designed and developed SIM-OTP application is also operated on GemXplore 3G V2 tool as shown in Figure 9. The test results showed that the OTP data is successfully verified by the developed OTP Validator (i.e., OTA server application).



(a) Security Parameters Indication (b) Key and algorithm Identifier for ciphering

Fig. 9. OTP operations of SIM-OTP module by GemXplore 3G V2

4. Conclusions

This paper presents the development and operation details of proposed SIM card based key management system as SIM-GAYS on SIM cards. The functional requirements regarding asymmetric key operations, symmetric key operation, signing operations and OTP operations on SIM-GAYS framework were developed and operated on GemXpresso JCard Manager. The results showed that the designed and developed APDU (Application Protocol Data Unit) transmission method between SIM Card and smartphone, and the Master Key based secure storage of keys support almost all cryptographic services provided by SIM-GAYS framework. The further work will focus on the validation and integration of the system with real-world service applications.

5. Acknowledge

This work is funded by Turkcell Technology A.S. and TUBITAK (The Scientific and Technological Research Council of Turkey) under 1505 Program, Project no 5180027.

References

3GPP (1999a). GSM 11.14-SIM Application Toolkit (SAT) for the Subscriber Identity Module.

3GPP (1999b). GSM 3.48-Security Mechanisms for SIM Application Toolkit.

Alliance, S. C. (2016). Smart card standards and specifications. Retrieved from http://www.smartcardalliance.org/smart-cards-intro- standards/.

Borst, J., Preneel, B., & Rijmen, V. (2001). Cryptography on smart cards. Computer Networks, 36(4), 423-435.

Ok, K., Coskun, V., Cevikbas, C., & Ozdenizci, B. (2015, November). Design of a key exchange protocol between SIM card and service provider. In 2015 23rd Telecommunications Forum Telfor (TELFOR) (pp. 281-284). IEEE.

Ok, K., Coskun, V., Yarman, S. B., Cevikbas, C., & Ozdenizci, B. (2016). SIMSec: A key exchange protocol between SIM card and service provider. Wireless Personal Communications, 89(4), 1371-1390.

Ozdenizci Kose, B., Cevikbas, C., Mantar, H.A., Buk, O., Coskun, V. (2020, September). Design of a Secure Key Management System for SIM Cards: SIM-GAYS. In Proceedings of 5th International Conference on Computer Science and Engineering (UBMK'20). IEEE.

Ozdenizci Kose, B., Morkoyun, S.E., Alsadi, M., Mantar, H.A., Coskun, V. (2019, October). A SIM Card Based Key Management System. In Proceedings of International Conference on Advances in Business Management and Information Technology (ICABMIT'19).