ACTA INFOLOGICA 2020;4(1):21-33
dergipark.org.tr/acin

DOI: 10.26650/acin.552116

RESEARCH ARTICLE

Determining the Cryptography Algorithm and Model for Mobile Payment Systems

Mobil Ödeme Sistemleri için Şifreleme Algoritmasının ve Modelinin Belirlenmesi

Öznur Şengel[1,2], Muhammed Ali Aydın[2], Ahmet Sertbaş[2]

ÖZ

Mobil ödeme sistemi son zamanlarda en yeni ve en popüler teknoloji olmaktadır. Mobil ödeme sistemi kredi kartı bilgileri olmaksızın hızlı ve güvenli ödeme kanalı sağlayan bir uygulamadır. Tüm ödemeler ya tanımlı olan operatör hattının faturasından ya da telefondaki uygulama hesabından yapılabilmektedir. Her uygulamanın farklı gereksinimleri vardır. Mobil ödeme sistemlerinin ana gereksinimleri fonksiyonellik, güvenlik ve hızdır. Mobile ödeme sistemlerinde ödeme esnasında en önemlisi hız etkenidir. Eğer güvenlik uygulamada en az zaman tüketimini sağlamıyorsa bu sistem tercih edilmemektedir. Bu yüzden mobil ödeme uygulamalarına en uygun modeli ve algoritmayı belirlemek için bu çalışmada şifreleme algoritmalarının zaman tüketimini kontrol ettik. Bu çalışmada mobil ödeme sistemleri için en uygun şifreleme modeli ve algoritmayı bulmaya çalışmaktayız. En çok bilinen asimetrik anahtarlı şifreleme olan Rivest-Sahmir-Adleman ile en çok bilinen simetrik algoritmaları olan Veri Şifreleme Standardı, Üçlü Veri Şifreleme Standardı, Geliştirilmiş Şifreleme Standardını şifreleme ve deşifreleme işlemleri esnasında tükettikleri zamanlara göre karşılaştırdık. Çalışmanın sonucu olarak Geliştirilmiş Şifreleme Standardı diğer algoritmalardan yaklaşık olarak üç kat daha hızlı olduğu gözlenmiştir.

Anahtar kelimeler: Mobil Ödeme Sistemi, Şifreleme Algoritması, Şifreleme Modeli

ABSTRACT

Mobile payment systems are becoming one of the most popular technologies nowadays. A mobile payment system is an application that provides a payment channel easily and quickly without credit card information. All payments can be either over GSM bill or from your phone application account. Each mobile application has different requirements. The main requirements of mobile payment systems are functionality, security and speed. The cryptography model and algorithm are very important to make all transactions securely on mobile payment applications. The speed factor is also very important during payment on mobile payment applications. If security does not provide a minimum time consumption on application, this system becomes not preferable. Therefore, we analyzed the time consumption of the cryptographic algorithms to specify the best model and algorithm for mobile payment applications. In this study, we tried to find most suitable cryptographic model and algorithm for mobile payment systems. We compared Rivest-Shamir-Adleman, which is a well-known asymmetric key algorithm, with well-known symmetric algorithms such as Data Encryption Standard, Triple Data Encryption Standard, and Advanced Encryption Standard in terms of time consumption of the algorithm over encryption and decryption processes. As a result of this study, Advanced Encryption Standard was found to be approximately three times fast than among all algorithms.

Keywords: Mobile Payment System, Cryptography Algorithm, Cryptography Model

[1]İstanbul Kültür University, Department of Computer Engineering, İstanbul, Turkey
[2]İstanbul University-Cerrahpaşa, Department of Computer Engineering, İstanbul, Turkey

ORCID: Ö.Ş. 0000-0002-2186-927X;
M.A.A. 0000-0002-1846-6090;
A.S. 0000-0001-8166-1211

Corresponding author:
Öznur Şengel,
İstanbul Kültür University, Computer Engineering, İstanbul, Turkey
Telephone: +90 212 498 47 20
E-mail address: o.sengel@iku.edu.tr

Submitted: 10.04.2019
Revision Requested: 07.10.2019
Last Revision Received: 11.11.2019
Accepted: 06.03.2020

Citation: Şengel, Ö., Aydın, M.A. & Sertbaş, A. (2019). Determining the cryptography algorithm and model for mobile payment systems. *Acta Infologica, 4*(1), 21-33. https://doi.org/10.26650/acin.552116

1. INTRODUCTION

Developments in technology change with lives and abilities. Today's generation wants to reach everything easily and quickly. Therefore, everybody prefers to use mobile applications. This new trend for shopping goes on mobile applications. There are many applications for shopping on a mobile phone, such as an online wallet and mobile wallet. Some of these applications are more suitable than using a credit card and some mobile applications are suitable for near field communication on payment.

Mobile devices can be used for payment instead of giving credit card information, with near field communication (NFC). Credit card information is stored either in a mobile application or in the server of the applications, so the security of application is very important. Security of the system in a mobile application is related with security of data whereas information is transmitted between the layers. Security algorithms are used for security of data in applications. Credit cards have both hardware and software security options. On the other hand, mobile applications have only software security options.

Security of the system depends on choice of the most suitable cryptographic model and algorithm. Cryptography model means which information of the system is reached by the adversary. There are three cryptography models; black box cryptography model, gray box cryptography model, and white box cryptography model. Cryptography algorithm defines both encryption and decryption process of the system. There are two kinds of cryptography algorithms according to key types: private key cryptography and public key cryptography.

In this paper, we compared cryptography models and algorithms for mobile payment systems. There are lots of criteria to find the best model and algorithm for mobile applications such as security, battery consumption, time consumption, attack resistance, storage consumption, hardware/software suitability (Mahajan & Sachdeva, 2013; Mathur & Kesarwani, 2013; Padmavathi & Kumari, 2013; Singhal & Singhal, 2016).

The most important criterion is speed of all processes. We tested our system with Data Encryption Standard (DES) that was published in 1993 by the Federal Information Processing Standards Publication, Triple Data Encryption Standard (Triple DES) (Barker & Mouha, 2017), Advanced Encryption Standard (AES) was published by Federal Information Processing Standards Publication in 2001, and Rivest-Shamir-Adleman (RSA) (Jonsson & Kaliski, 2003) algorithms to specify which one has high speed during the encryption and decryption processes.

The second important criterion is robustness of the system against attacks. The resistance of the system is based on first the cryptography model, then the algorithm. The features of the cryptography models prepare the system environment such as input, output, and security algorithm. Black box cryptography model, gray box cryptography model, and white box cryptography model have different properties for the system environment.

The paper is organized as follows. Section 2 gives information about the mobile payment systems working schema. Section 3 deals with cryptography, cryptography models, and cryptography algorithms. Section 4 shows which cryptography algorithm and model are suitable for mobile payment systems and shows the results of the comparison.

2. MOBILE PAYMENT SYSTEMS

All over the world, a mobile device can be used as an online wallet, mobile wallet, and short message services (SMS) based mobile payment. Payment tools are specified by mobile operators, instead of banks. Mobile payment applications need some information to make payments properly. Mobile payment system ensures quick and easy payment operations without cash and credit card information. When you use mobile payment apps, payment is done over a mobile phone line of the global system for mobile communications (GSM).

Mobile payment systems use either proximity payments or mobile remote pay, but some of them use both of these methods if the device has all the required features. Mobile remote pay method does not need to use secure elements because it uses different authentication while using system. These systems use sim cards to approve the payment by SMS. A system with mobile remote pay method uses an authentication payment service provider. On the other hand, systems with proximity payment must have NFC, secure element and interfaces. As shown in Fig. 1, the host controller provides communication

between the Mobile Network Infrastructure and the host interfaces. A secure element makes this communication securely while the NFC controller sends user information over communication channels.

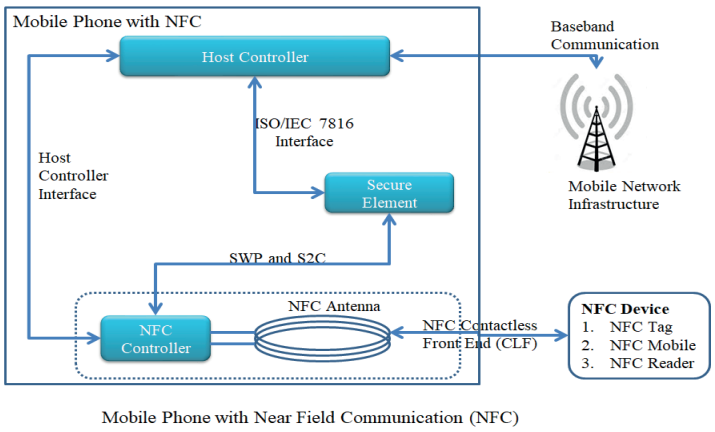


Figure 1. Mobile phone structure with NFC

NFC is a short distance, non-contact technology standard. It is designed for easy, simple, and secure communication between two electronic devices. Mobile phone applications can store more than one card, such as a debit card or a credit card, so that you can pay via the contactless terminal on your mobile phone instead of giving a card from your wallet to the cashier during payment. NFC has interactivity, remote multi-application management, and remote user management features. Interactivity means that users can use phone functions such as screen, vibration, voice etc. to use NFC services. The advantages of downloading, personalization, and opening / blocking applications are provided by remote multi-application management used in contactless cards. Service providers can access NFC service usage records and send personal information by user permission with remote user management functions.

The current security measures of payment systems also apply to NFC-compatible mobile phones. If you lose your bank card or credit card, the procedure is the same as if you lose your mobile phone with a payment card. You will be able to prevent the usage of your mobile phone's payment feature with Over-the-Air (OTA).

There are ongoing studies against NFC's malicious attack in the literature. The usage of encrypted communication and nesting mechanisms ensure the creation of a secure communication channel for blocking the connection between two NFC devices. Encrypting and storing all data in a different way from cryptographic mechanism provides secure communication against fraud. NFC payment infrastructure is developing by preventing new attacks with special processes. After the initial fraud attempt, usage of the card is blocked by creating blacklists and card identification keys are not created from the master key against cloning.

3. CRYPTOGRAPHY

Cryptography is a kind of cipher science concerned with reliable data communication. It is used to prevent the usage or modification of various messages by third parties when the information is transmitted in public environments. Cryptography has been used for a long time to deliver information safely to the target person. The oldest encryption methods are known as permutation and substitution. Briefly, permutation is done by changing the position of letters in a text, substitution is done by replacing the letters with other letters in a text.

The data, that is plaintext, is sent over an unsecure network, so data can be eavesdropped and modified by a third person/party. Plaintext is encrypted with a mathematical algorithm to generate a hidden message, which is called ciphertext. If someone reads the ciphertext without an algorithm, cannot understand anything. Nowadays, the algorithms used in applications are not hidden.

The security of the knowledge is not related with a hidden algorithm. The most important part is hiding and securing the key that is used in the algorithm during the encryption process. Different keys can be generated, only the person who knows the key correctly can obtain the plaintext from the ciphertext. Therefore, the key must be unrepeatable and unique in encryption systems.

As shown in Fig. 2, Computer A wants to send an important message to Computer B over an unsecure channel. Computer A uses the encryption algorithm to generate a ciphertext and sends it to Computer B. Computer B recovers plaintext from the received ciphertext.

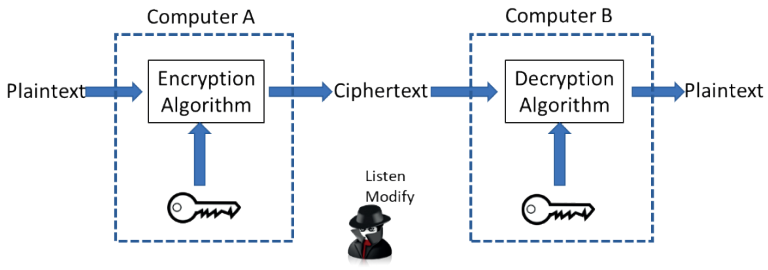


Figure 2. Cryptography schema

3.1. Cryptography Models

There are various attacks according to the cryptographic model. Each model has different a working process so the attacker catches some information during the execution of the algorithm in this model.

Black Box Cryptography

The attacker has no information about how the algorithm works, how the key is used, which process exists, etc. By the way, they do not know internal processes and do not have access to the key. They only know external information. The third users can obtain only input (plaintext) and output (ciphertext) (Fig. 3). A system with a black box model does not allow to obtain execution code, encryption and decryption processes, and the key generation operations. Although the third parties know the input and output of the algorithm, they do not know details of the encryption and decryption processes to execute the system.

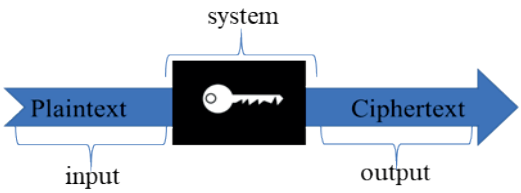


Figure 3. Black box cryptography model

Gray Box Cryptography

In the gray box model, the attacker gets more information than black box model. This model allows that the adversary can observe side channel information such as power consumption, timing information, electromagnetic radiations, and fault analysis of system (Fig. 4). They use this information to obtain plaintext of the system. When the encryption algorithm runs, some analysis information can be observed, such as power analysis. According to this information, attackers can detect some important points of the system. The peak values of analysis give clues about the operation of the algorithm in the system.

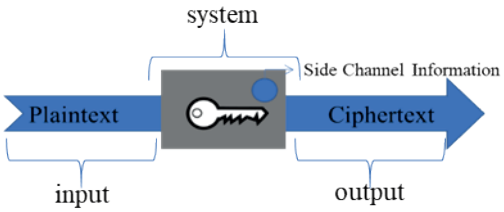


Figure 4. Gray box cryptography model

White Box Cryptography

The White Box Cryptography (WBC) (Beunardeau, Connolly, Geraud, & Naccache, 2016) aims to protect the key in obfuscated cryptographic implementation. In a system with this model, everything can be observed such as input, output, intermediate calculation in algorithm, and memory visibility. The attackers have full control in the dynamic execution as shown in Fig. 5 so they can obtain important data such as the key. Since the algorithm can be observed, it can also be altered.

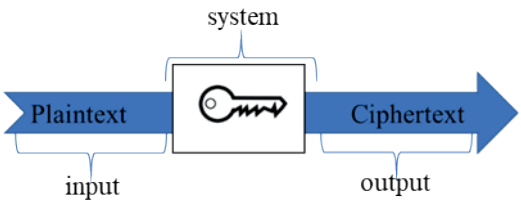


Figure 5. White box cryptography model

In the literature, the first application on white box cryptography started with Chow et al.'s study on white box cryptography with AES (Chow, Eisen, Johnson, & Oorschot, 2003b), and with DES (Chow, Eisen, Johnson, & Oorschot, 2003a). Both of them have the same structure of allocation to the loop function on the encryption while using a small size lookup table. These two structures AES and DES were broken by Wyseuret et al. (2007) with 2^{14} complexity and by Lepoint et al. (2014) with 2^{22} complexity respectively. The CEJO structure has been used by most of the researchers either to develop existing structure or to break (Billet, Gilbert, & Ech-Chatbi, 2004; Michiels, Gorissen, & Hollmann, 2009) the existing system. Some researchers (Delerablée, Lepoint, Paillier, & Rivain, 2014; Saxena, Wyseur, & Preneel 2009) studied security notations of white box cryptography such as unbreakability, one-wayness, incompressibility, and traceability according to attack scenarios. If the application process time is important, instead of authentications limits, the white box cryptography model is the best one (Şengel, Aydin, & Sertbaş, 2018).

3.2. Cryptography Algorithms

Private key cryptography

Private key cryptography is known as symmetric cryptography or secret key cryptography that uses a single key for both encryption and decryption. Both sender and receiver must have the same key in the symmetric cryptography algorithms (Fig. 6). If the attacker received the ciphertext on a communication channel, the message would not be read without the secret key.

Private key cryptography is used for not only encryption but also for authentication. Message Authentication Codes (MAC) are used for authentication and digital signatures. They use only one agreed key instead of two different keys. So, they do not need verifications.

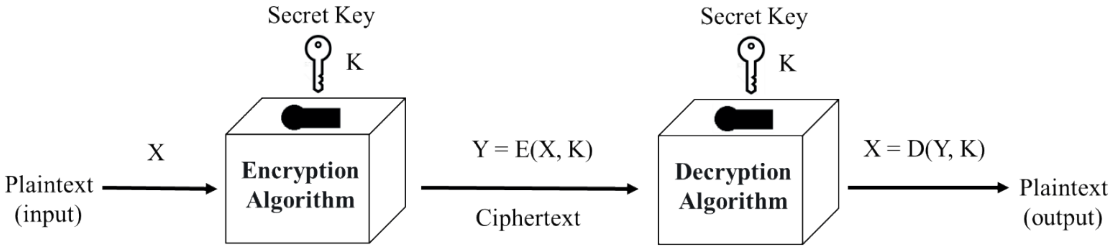


Figure 6. Private key cryptography encryption and decryption

The basic problem of private key cryptography is that the sender and the receiver agree on a key without taking possession of it by attackers. The aim of cryptography is to resist the third user to find the secret key. Key agreement protocols are used to specify the key. Caesar, Monoalphabetic, DES, Triple-DES, RC5, Blowfish, CAST-128, IRON, and AES are well known symmetric key algorithms.

Data Encryption Standard (Federal Information Processing Standards Publication, 1993)

At the end of the 1960s, a group of researchers under Horst Feistel in IBM, developed a cryptographic system that is named LUCIFER, and is used in the USA. In 1973, the US standards institute NIST (National Institute of Standards and Technology) invited companies to establish a standard for civil usage. As a result of these investigations, the closest solution was found to be LUCIFER. The US Security Agency (NSA) specialists worked on LUCIFER with a 128-bit password key, they made some adjustments and reduced the key length to 56-bit. This new algorithm was published as DES in 1977, and started to be used as a standard in many areas, particularly in the finance industry.

DES algorithm is a symmetric encryption algorithm that uses private key management. DES uses the Feistel structure for encryption as many other symmetric encryption algorithms. The Feistel structure is a helical structure. Text is divided into two parts, and the operation is performed on only one of them at each stage, and the second half of the data on the second stage.

DES has a mythical place in classical encryption systems, and even today, it forms the cryptographic backbone of all card systems such as VISA, MASTERCARD, BKM, etc. DES is designed to perform the mixing and replacement processes with extreme care and systematically.

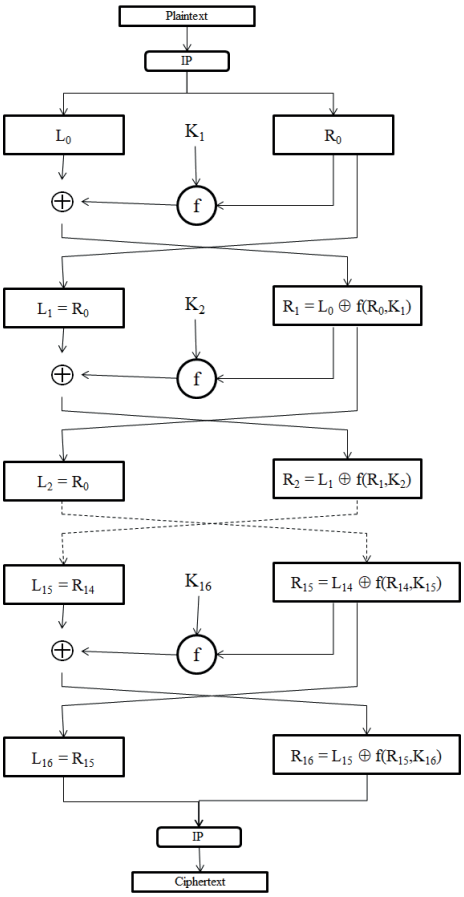


Figure 7. DES encryption structure

As show in Fig. 7, plaintext is applied to the initial permutation to separate text into two parts as left and right. The right part (R_0) is used as the left part (L_1) in the second round. The right part (R_0) applies f function with the key (K_1) of the algorithm. The result of the function applies exclusive-or (XOR) with the left part (L_0) to generate the right part (R_1) of the next round. The DES algorithm has 16 rounds, each round is applied to the same functions. Finally, the algorithm gets ciphertext with the left part and the right part of the last round, and they combine with the initial permutation.

Triple Data Encryption Standard

Triple DES (Barker & Mouha, 2017) is an encryption algorithm developed by IBM in 1978. It was developed on DES algorithm, which is difficult, to resist Brute Force attacks. Triple DES is an encryption technique that is created by the successive operation of the Standard DES, with two or three keys of 112 bit.

As shown in Fig. 8, 128-bit keys are divided into two 64-bit parts. The first part of the key is used in the first and third DES, the second part of the key is used in the second DES. Triple DES is commonly used in bank systems, electronic payment systems, and to generate a software key.

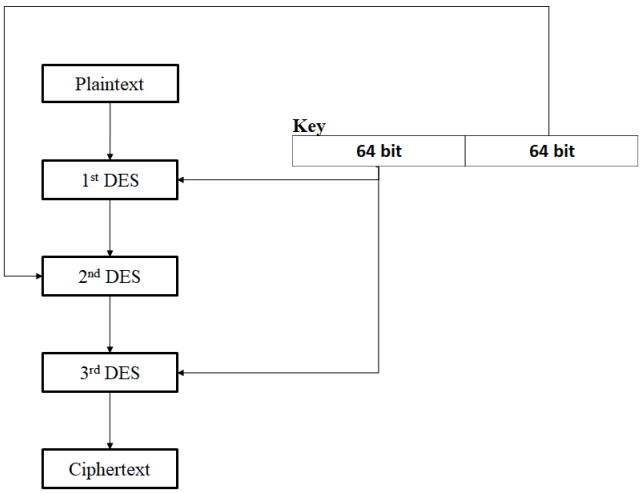


Figure 8. Triple-DES working schema

Advanced Encryption Standard (Federal Information Processing Standards Publication, 2001)

AES was announced by the US National Institute of Standards and Technology (NIST) on November 26, 2001, with US FIPS PUB 197 document. Standardization was completed over a period of five years. In this process, 15 designs were presented as AES nominees. After the evaluation of nominee designs in terms of security and performance, the most appropriate design was chosen as the standard encryption algorithm. AES is based on the Rijndael algorithm, developed by Vincent Rijmen and Joan Daemen. Rijndael is obtained by using developers' names: RIJmen aNd DAEmen.

The encryption algorithm defined as AES is a symmetric key algorithm in which both the encryption and decryption keys are related. AES is based on Substitution-Permutation. There are three versions of AES according to key sizes that are 128-bit, 192-bit, and 256-bit. For each version, AES uses different round numbers. AES-128 uses 10 rounds, AES-192 uses 12 rounds and AES-256 uses 14 rounds. Each round, except the last round, includes four sub processes: sub bytes, shift rows, mix column, and add round key.

SubBytes: The value of each byte in the state matrix is updated by using an 8-bit substitution box. This step disrupts the linearity of the algorithm and makes a non-linear transformation. It is obtained from the inverse operation on the finite field GF (2^8) of the substitution box that is known to have high nonlinearity. In order to be resistant against attacks by using algebraic properties, a further linear inversion is added to the inverse operation on the finite object.

ShiftRows: This process runs on the rows of the matrix and shifts the bytes in each row to the left with a certain number value. The first row remains constant both in AES-128 and in AES-192, while the 2nd, 3rd, and 4th rows are shifted left by 1, 2, and 3 bytes respectively. The first row remains constant in the Rijndael algorithm for 256-bit, while the 2nd, 3rd, and 4th rows are shifted left by 1, 3, and 4 bytes respectively.

MixColumn: The four bytes in each column are mixed with each other using a linear transformation. The MixColumn function takes 4 bytes of input and gives 4 bytes of output. This step ensures that each byte in the input affects each byte value in the output. This process consists of multiplying each column with a fixed matrix. The matrix multiplication operation is performed on the finite field GF (2^8). The constant matrix in the MixColumn step is an MDS matrix and provides a complex diffusion with the ShiftRows step.

AddRoundKey: In each round, the algorithm generates a new round key. The new round key applies exclusive or operation with a state matrix.

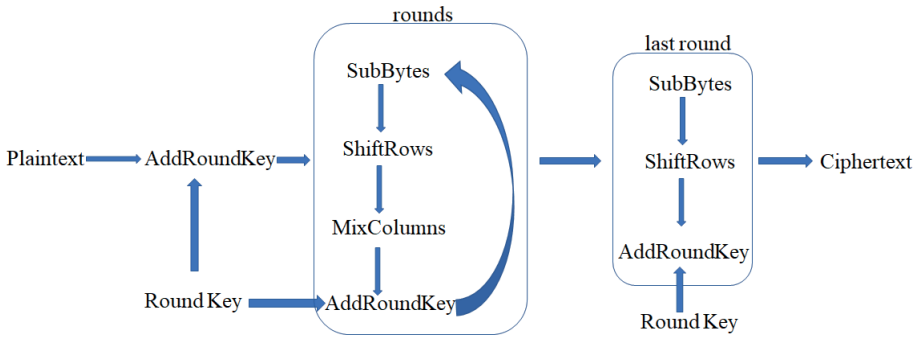


Figure 9. AES encryption schema

As shown in Fig. 9, first of all, the key expansion process uses a key to generate round keys. In the first round, state that is 4 x 4 matrix, applies XOR with the firstround key. The other rounds include SubBytes, ShiftRows, MixColumn, and AddRoundKey steps respectively. The last round does not include the MixColumn process.

Public key cryptography

Public key cryptography is known as asymmetric cryptography that uses two different keys for encryption and decryption. One of these keys is a public key, the other is a private key. The public key is shared with others for communication and everyone can reach it. On the other hand, the private key is known by owner. The two keys have a mathematical relationship, but generating one key from the other key is too difficult.

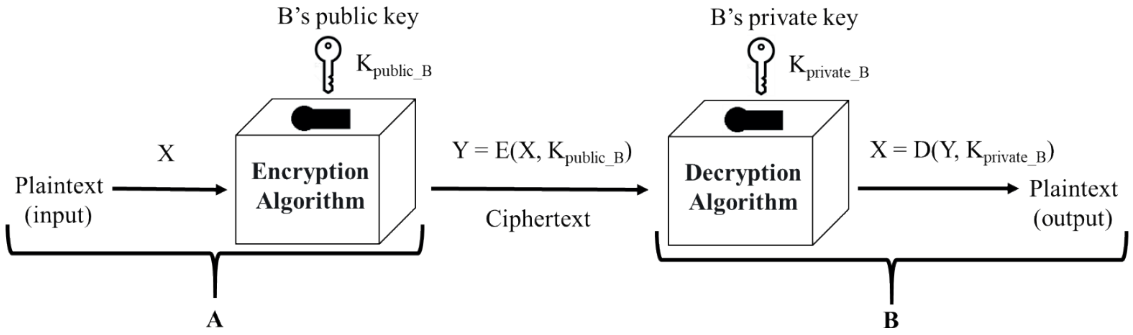


Figure 10. Public key cryptography encryption and decryption

As shown in Fig. 10, Computer A sends data to Computer B. Computer A encrypts data with the public key and generates ciphertext. Computer A sends this ciphertext to the Computer B. Computer B decrypts the ciphertext with the private key of Computer A. The most important part of this scenario is the authentication.

Rivest-Shamir-Adleman (Jonsson & Kaliski, 2003)

RSA is a public key cryptographic structure that allows both encryption and digital signature. In 1974, Ronald Rivest, Adi Shamir and Leonard Adleman, who used Public Key Cryptography management of Diffie and Hellman, revolutionized the RSA algorithm. This method, which works out of seemingly simple mathematical relationships, has two separate keys. One of the keys is open to the public, the other is available only to the owner. Everyone broadcasts the public key. Someone encrypts and sends the message by using this public key when he/she wants to

send an encrypted message. However, the message only can be decrypted with the secret key, which is the pair of the public key.

First, two prime numbers, *p* and *q* are chosen. *n* is obtained by multiplying the prime numbers *p* and *q*. *m* is obtained by multiplying *(p - 1)(q - 1).* We need to select an appropriate encryption key (*e*) that will be smaller than *n* and must be a prime relative to *m*. We need to find a decryption key (*d*) that allows *d* x *e*-1 number to be fully divided into *m,* and it must be less than *m*. *(e, n)* is the public key, *(n, d)* is the private key.

The most important part of RSA is that the prime number must be bigger for this algorithm. The *p* and *q* numbers should be more than 100 digits, and the *n* numbers should be more than ten thousand digits. Therefore, the RSA algorithm is 1000 times slower than the DES algorithm.

4. FINDINGS

All private key cryptography algorithms use data in blocks. DES and Triple-DES separate data into two blocks, whereas AES treats data as a single block. DES and Triple-DES are based on the Feistel encryption, AES is based on substitution and permutation, RSA is based on large prime integer numbers. The key size is very important for cryptography algorithms, because developing a more secure system is related with the key size. AES has different key sizes and RSA has the longest key size in all the cryptography algorithms. As seen in Table 1, security factors are related with the key size of the algorithms.

Table 1
Comparison of three important factors to develop a mobile payment system

Factors	DES	Triple-DES	AES	RSA
Security	key size is not enough	more secure than DES	related with key size	related with big prime numbers
Speed	slow	more slowly	fast	more slowly
Power consumption	minimum	maximum	minimum	maximum

Mobile payment systems include hardware or software applications. Therefore, the algorithm must be suitable for both hardware and software. When we compare the cryptography algorithms; DES and Triple-DES are used for hardware applications, RSA is not efficient for both hardware and software applications, but AES is used for both hardware and software applications. There are five important factors to consider while developing a mobile payment system: speed, security, power consumption, time consumption, and crypto analysis. As seen in Table 2, all of the cryptography algorithms have some weak parts, but AES is more robust against attacks.

Table 2
Crypto analysis of cryptography algorithms

Crypto analysis	Cryptography algorithms			
	DES	Triple-DES	AES	RSA
Weak against differential crypto analysis	ü	ü	x	x
Weak against linear crypto analysis	ü	x	x	x
Weak substitution table	ü	x	x	x
Weak against brute force attacks	x	ü	x	ü
Weak against differential crypto attacks	x	ü	x	x
Weak against oracle attacks	x	x	x	ü
Robust truncated differential, interpolation, square attacks	x	x	ü	x

We compared DES, Triple-DES, AES, and RSA in terms of time consumption. Time consumption of the system is the most important criterium because mobile systems must be as easy and fast as possible for payment. Mobile payment applications are running on Android based mobile phones. The application has wallet user interface, secure channel, NFC controller, and Hardware Security Module (HSM) library modules. The user interfaces of the mobile application are developed for active communication with users. The HSM library contains all the necessary components to enable the application to run on android devices. The secure channel module is a sub-library that ensures the communication between mobile payment applications and HCE cloud systems. The NFC Controller module enables communication between the pos device and the

application. The most important part of the system is to protect personal information so more secure and faster algorithms should be chosen. We tested all these algorithms on NetBeans IDE 8 with c programming language. Same input data that is the password of the payment app, (such as 1234 password of mobile payment) is used for all algorithms on the same platform.

We compared encryption time consumption as seen in Table 3. The average time consumption of the RSA, DES, AES-128, AES-192, AES-256 are 4600 milliseconds, 2000 milliseconds, 480 milliseconds, 589 milliseconds, 555 milliseconds respectively. Decryption of the algorithms gave approximately the same results. According to these results in Table 3, time consumption of private key cryptography algorithms is more or less than public key cryptography algorithms. As seen in Fig. 11, the AES algorithm is more efficient in terms of time consumption and AES has different time consumption values according to its key length. AES-128 is more efficient than other AES algorithms. Therefore, the AES algorithm is most suitable to use for mobile payment applications, especially AES with 128-bit.

Table 3
Test result of cryptography algorithms (in milliseconds)

Algorithm	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Average
RSA	8000	6000	5000	5000	5000	3000	3000	3000	4000	4000	4600
DES	5000	3000	2000	1000	2000	2000	1000	1000	1000	2000	2000
AES-128	828	375	374	374	437	377	376	828	453	375	480
AES-192	2000	375	438	828	375	375	375	375	375	374	589
AES-256	2000	375	437	375	438	375	429	375	375	374	555

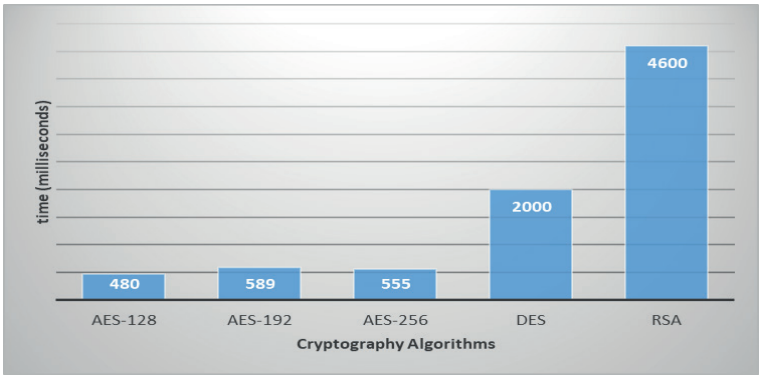


Figure 11. Average result in milliseconds for cryptography algorithms

As a result of the cryptography algorithm for encryption, AES is better than other algorithms we compared. Encryption and decryption time consumption of AES-128, AES-192, and AES-256 with the white box cryptography model is seen in Fig. 12. The time consumption of AES-128, AES-192, and AES-256 in encryption process are 597 milliseconds, 667 milliseconds, and 609 milliseconds respectively. The time consumption of AES-128, AES-192, and AES-256 in decryption process are 557 milliseconds, 578 milliseconds, and 769 milliseconds respectively.

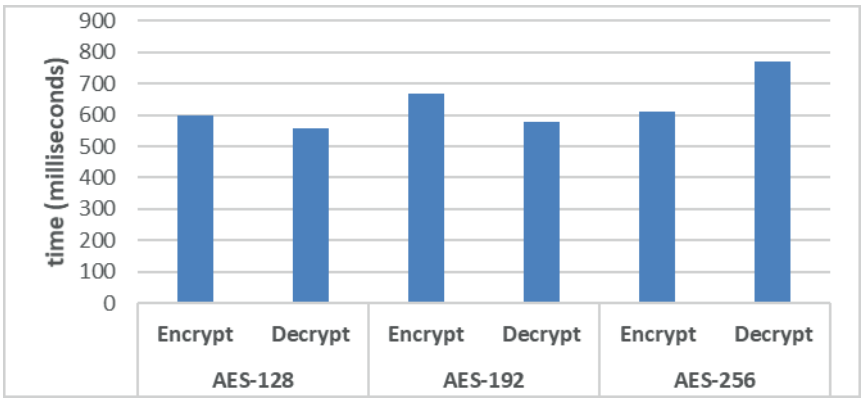


Figure 12. Result of encryption and decryption

5. DISCUSSION AND CONCLUSION

Several research studies show that third users usually attack to obtain the key in a system, so the key is the most important part of a cryptography algorithm. Public key cryptography seems more secure than private key cryptography because the key is not sent via a communication channel. On the other hand, public key cryptography is slower than private key cryptography. Public key cryptography is not replaced with private key cryptography because private key cryptography is used to make system stronger. For example, it is used to transmit a secret key via an unsecure communication channel.

In mobile payment systems, time is more important than security, so private key cryptography algorithms are more suitable than public key cryptography algorithms. Even though the attacker has full privilege with system environments, the white box cryptography model has less execution time than the other cryptography models. Unbreakability, onewayness, traceability, and incompressibility notions must be considered to construct more secure systems with the white box model.

Well-designed mobile payment systems must be constructed on the white box cryptography model with a strong private key cryptography algorithm, such as AES. The system will be faster and more secure with this cryptography model-algorithm pair. Although the system is reachable, the cryptography algorithm provides a more secure system.

Acknowledgement: This work is a part of the Ph.D. thesis titled "Model Design and Performance Analysis for Secure Storage of Personal Data in Mobile Payment Systems" at Institute of Graduate Studies, Istanbul University - Cerrahpaşa, Istanbul, Turkey.

Peer-review: Externally peer-reviewed.
Conflict of Interest: The authors have no conflict of interest to declare.
Grant Support: The authors declared that this study has received no financial support.

References

Barker, E., & Mouha, N. (2017). *Recommendation for the Triple Data Encryption Algorithm (TDEA) block cipher.* Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf

Beunardeau, M., Connolly, A., Geraud, R., & Naccache, D. (2016). White-box cryptography: Security in an insecure environment. *IEEE Security & Privacy*, *14*(5), 88–92. http://dx.doi.org/10.1109/msp.2016.100

Billet, O., Gilbert, H., & Ech-Chatbi, C. (2004). Cryptanalysis of a White box AES implementation. *Selected Areas in Cryptography Lecture Notes in Computer Science*, 3357, 227–240. http://dx.doi.org/10.1007/978-3-540-30564-4_16

Chow, S., Eisen, P., Johnson, H., & Oorschot, P. C. (2003a). A White-box DES implementation for DRM applications. *Lecture Notes in Computer Science Digital Rights Management*, 2696, 1–15. http://dx.doi.org/10.1007/978-3-540-44993-5_1

Chow, S., Eisen, P., Johnson, H., & Oorschot, P. C. (2003b). White-box cryptography and an AES implementation. *Selected Areas in Cryptography Lecture Notes in Computer Science*, 2595, 250–270. http://dx.doi.org/10.1007/3-540-36492-7_17

Delerablée, C., Lepoint, T., Paillier, P., & Rivain, M. (2014). White-Box Security Notions for Symmetric Encryption Schemes. *Selected Areas in Cryptography - SAC 2013 Lecture Notes in Computer Science*, 8282, 247–264. http://dx.doi.org/10.1007/978-3-662-43414-7_13

Federal Information Processing Standards Publication: Advanced encryption standard (AES). (2001). http://dx.doi.org/10.6028/nist.fips.197

Federal Information Processing Standards Publication: Data encryption standard (DES). (1993). http://dx.doi.org/10.6028/nist.fips.46-2

Jonsson, J., & Kaliski, B. (2003). *Public-Key Cryptography Standards (PKCS)#1: RSA cryptography specifications version 2.1* (pp. 1–68). http://dx.doi.org/10.17487/rfc3447

Lepoint, T., Rivain, M., Mulder, Y. D., Roelse, P., & Preneel, B. (2014). Two attacks on a white-box AES implementation. *Selected Areas in Cryptography Lecture Notes in Computer Science*, 8282, 265–285. http://dx.doi.org/10.1007/978-3-662-43414-7_14

Mahajan, P., & Sachdeva, A. (2013). A study of encryption algorithms AES, DES and RSA for security. *Global Journal of Computer Science and Technology, 13*(15).

Mathur, M., & Kesarwani, A. (2013). Comparison between Des, 3des, Rc2, Rc6, Blowfish and Aes. In *Proceedings of National Conference on New Horizons in IT-NCNHIT* (Vol. 3, pp. 143–148).

Michiels, W., Gorissen, P., & Hollmann, H. D. (2009). Cryptanalysis of a generic class of white-box implementations. *Selected Areas in Cryptography Lecture Notes in Computer Science*, 5381, 414–428. http://dx.doi.org/10.1007/978-3-642-04159-4_27

Padmavathi, B., & Kumari, S. R. (2013). A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution. *International Journal of Science and Research (IJSR), India, 2*(4).

Saxena, A., Wyseur, B., & Preneel, B. (2009). Towards security notions for white-box cryptography. *Lecture Notes in Computer Science Information Security*, 49–58. http://dx.doi.org/10.1007/978-3-642-04474-8_4

Singhal, S., & Singhal, N. (2016). A Comparative Analysis of AES and RSA Algorithms. *International Journal of Scientific & Engineering Research, 7*(5), 149–151.

Şengel, Ö., Aydin, M. A., & Sertbaş, A. (2018). A survey on white box cryptography model for mobile payment systems. *Lecture Notes in Electrical Engineering International Telecommunications Conference*, 504, 215–225. http://dx.doi.org/10.1007/978-981-13-0408-8_18

Wyseur, B., Michiels, W., Gorissen, P., & Preneel, B. (2007). Cryptanalysis of white-box DES implementations with arbitrary external encodings. *Selected Areas in Cryptography Lecture Notes in Computer Science*, 264–277. http://dx.doi.org/10.1007/978-3-540-77360-3_17