# Comparison of Cryptography Algorithms for Mobile Payment Systems

Ö. ŞENGEL[1], M. A. AYDIN[2], A. SERTBAŞ[3]

[1] İstanbul Kültür University, İstanbul/Turkey, o.sengel@iku.edu.tr
[2] İstanbul University-Cerrahpaşa, İstanbul/Turkey, aydinali@istanbul.edu.tr
[3] İstanbul University-Cerrahpaşa, İstanbul/Turkey, asertbas@istanbul.edu.tr

***Abstract* – Mobile payment services are the newest and most popular technology that is developing according to our habits and needs. Consumer all over the world are using mobile phone for payment as well as communication. The main purpose of using mobile payment application is doing all transaction easily and quickly. Not only data security in electronic transactions, but also the speed of the system operations is becoming very important. There is a threshold value to finish all transaction in mobile payment systems. If the security algorithm is more complex and exceed threshold, it is not suitable to using in mobile payment systems. In this paper we compare cryptography algorithms and proposed two algorithms on Advanced Encryption Standards. The experiment results show that proposed algorithms is suitable cryptography algorithm for mobile system according to time and storage consumption factors.**

***Keywords* – Cryptography, Mobile Payment Systems, Cryptography Algorithms, Data Security.**

## I. Introduction

MOBILE payment systems are based on the encryption algorithms that used in debit cards. There are cryptography algorithms used as the standard by Bankalar arası Kart Merkezi (BKM) in Turkey. Data Encryption Standard are used to transfer cipher text to center for authentication. There is no any security improvement during transaction between user and their bank in the mobile payment systems. Security of mobile applications is based on the security of the data in the other layers of application.

Hardware Security Module are used in credit card for bank application. Hardware Security Module (HSM) is a hardware device designed to protect and manage sensitive cryptographic keys required for strong authentication, securely store in physical medium, and perform cryptographic operations in the fastest manner. In addition to having a lot of encryption per minute, HSM devices can do this very quickly because of their special design, which reduces the processor load to the minimum.

Today HSM is being used to minimize the occurrence of events such as increased fraud, phishing and stolen personal information. HSM has more advantages both on security and on performance. The key protection operations are carried out in the module and the protection is done with special root keys. HSM has the self-resetting feature against attack. Therefore, many applications strengthen software part and integrated with HSM to make their application difficult to be broken by third parties.

The paper is organized as follows. Section 2 deals with how the mobile payment systems work and important points of mobile payment applications. Section 3 gives information about what is cryptography and cryptography algorithms. Section 4 describes which cryptography algorithm is suitable for mobile payment system and proposed algorithms.

## II. Mobile Payment Systems

Mobile payment is a service that allows you to make payments easily and quickly without the need for credit card information or cash over the application you use on our mobile phone. Any service or product that we purchase with our mobile phone is paid by the GSM bill or from the balance that is defined on our phone line. In this case, we can define the mobile payment as a payment system, in which all payment data and transaction are transmitted by the approved acknowledgment receipt accepted by the mobile device.

Mobile payment systems, which are increasing all over the world, are confronted in our country as systems used by domestic and foreign companies as Online Wallets, Mobile Wallets, SMS Based Payment Systems [1]. Payment tools, which are usually renewed by banks, are now beginning to lead innovations offered by mobile operators in the case of mobile payment methods.

There are two methods for mobile payments: Proximity Payments and Mobile Remote Pay. If the mobile device has the required features, it is possible to make payments in both types. For example, a payment system using Mobile Remote Pay while using the text message service on the mobile phone, another system application may request to be installed on the device. It is not necessary to use a secure element in the Mobile Remote Payment model. Because the system is configured according to the method used by the authentication payment service provider, the consumer (the paying party) authenticates directly from the payment server or uses the security features found on the SIM cards. There must be a Secure Element as well as a Near Field Communication Controller and interfaces that guarantee the secure operation of the application on Proximity Payments (Figure 1). Proximity Payments is a software that fulfills the function of a payment card that can directly access Near Field Communication (NFC) and communicate with the transaction point.

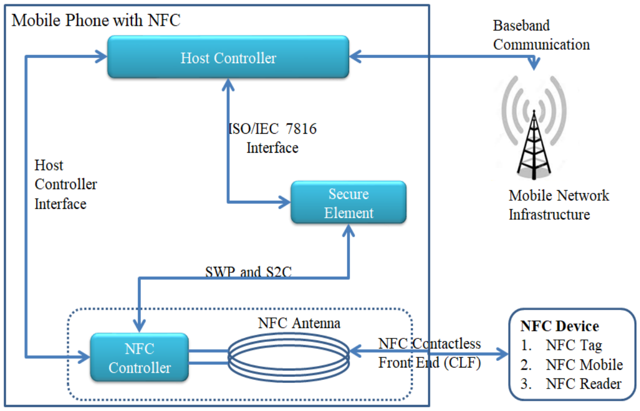

Figure 1: Mobile Phone with Near Field Communication (NFC)

Near Field Communication (NFC) Technology is a wireless technology that has been in the ISO RFID standard since 2003 and can be used for low-power data exchange, providing reliable access to short distance electronic devices and making reliable contactless transactions. When the customer wants to pay the payment through the mobile point of sale, he transmits the encrypted data by moving the mobile phone with the necessary hardware to the POS (Point of Sale) device. Payment is made after verifying the encrypted data transmitted.

RFID (Radio Frequency Identification) technology is a technology that uses radio waves to make mobile POS payments such as Near Field Communication Technology much longer and less secure than NFC technology in which RFID tags are read and transferred. Near Field Communication Technology is safer than RFID technology, but the security of communication between the buyer and the seller around the mobile payment center, such as secretly listening and watching the network to gain unfair earnings, gains importance.

## III. Cryptography

The encryption process is the whole process of preventing modification of the data during transmission. If we look at traditional cryptographic logic, it consists of two parts, encryption and decryption. The computer A wants to send the data to the computer B over a secure channel that an attacker can listen the network. Data with a secret key, that are known from computers A and B, are encrypted by computer A with an encryption algorithm. Encrypted data is sent to computer B over a secure channel. The computer B obtains plain text from the cipher text by decrypting it in the decryption algorithm with the secret key.

Encryption systems are divided into public key (asymmetric) and private key cryptography (symmetric) according to the key type used.

Two separate keys are used in public key cryptography (Figure 2): public key for encryption and private key for decryption. Everyone can know public key while the secret key can only be known in the person who decrypts it. Public key cryptography algorithms are used for digital signatures are used in the fields of authentication, information integrity and to securely identify and exchange the key to be used by the two parties. Deffie-Helman (DH), Rivest-Shamir-Adleman (RSA), ElGamal and Paillier are well-known public key cryptography algorithms.

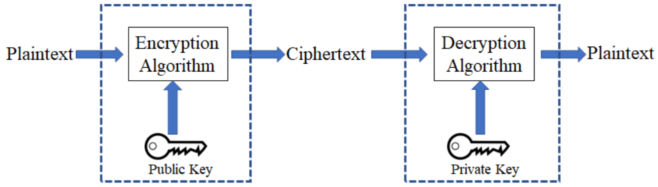

Figure 2: Public key cryptography

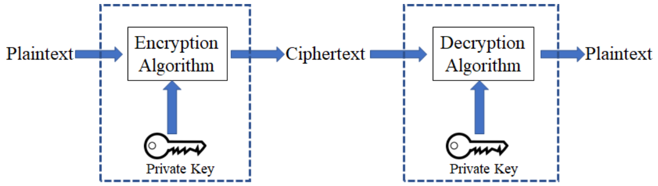

Figure 3: Private key cryptography

Private key cryptography (Figure 3) uses a single key for encryption and decryption. The sender also sends the agreed key along with the cipher text, so the key must be communicated securely to the destination. It is more secure than public key cryptography as long as the key is securely communicated to the other side. Caesar, Vigenere, Data Encryption Standard (DES), Triple DES (3DES), RC5, Blowfish, IDEA, SAFER, Advanced Encryption Standard (AES) are well-known private key cryptography algorithms.

### A. *Rivest-Shamir-Adleman*

Rivest-Shamir-Adleman (RSA) algorithm developed in 1977 by RON Rivesti Adi Shmir and Leonard Adleman. It is most widely used public key cryptography algorithm in digital signature. The security of the algorithms based on difficulty of large integers.

RSA [2] uses mathematically linked keys; public and private. Public key is different form private key and shared with everyone. The most complex part of RSA is key generation algorithm. n is calculated by multiply two large prime numbers p and q. Totient is calculated by multiply one minus of p and q. According to this calculation key pair (d, e) is selected.

Plaintext (P) is encrypted to ciphertext (C) by C = Pe mod n. the plaintext is produced by P = Cd mod n.

### B. *Data Encryption Standard*

In 1973, the National Bureau of Standards was a reconstruction of an earlier cryptographic system known as LUCIFER, developed by IBM when a request for cryptographic systems in the Federal Register was found. Data Encryption Standards (DES) [3] is based on data Encryption Algorithm.

A 64-bit plain text is encrypted with a 56-bit key and generate 64-bit cipher text by using data encryption algorithm (Figure 4). DES is a block encryption algorithm based on symmetric encryption principle. The same algorithm and key are used both encrypt and decrypt data blocks.

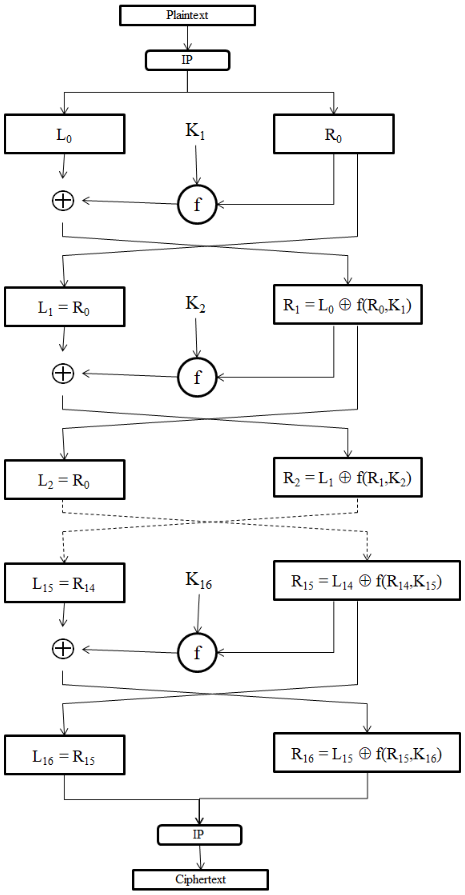
Plaintext
IP
$L_0$
$K_1$
$R_0$
f
$L_1 = R_0$
$K_2$
$R_1 = L_0 \oplus f(R_0,K_1)$
f
$L_2 = R_0$
$R_2 = L_1 \oplus f(R_1,K_2)$
$L_{15} = R_{14}$
$K_{16}$
$R_{15} = L_{14} \oplus f(R_{14},K_{15})$
f
$L_{16} = R_{15}$
$R_{16} = L_{15} \oplus f(R_{15},K_{16})$
IP
Ciphertext

Figure 4: Data Encryption Standard encryption schema

As shown in Figure 4, plain text (x) separates two 32-bit block (L0 and R0) with fixed initial permutation in (1). After first operation, there is operation sequence 16 times. In each step, Li is generated by (2) and Ri is generated by (3) by adding key (K).

$$x_0 = IP(x) = L_0R_0 \tag{1}$$

$$L_i = L_{i-1} \tag{2}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \tag{3}$$

In Equation (3), Ri-1 expands to 48-bit data with expansion permutation function. This 48-bit data exor with 48-bit key that selected from 56-bit original key. After operation 8 blocks are generated (Bj, j=1,…,8) and from each blocks 6-bit values generates 8 s-box. Finally, 32-bit result is generated from s-box and each s-box is produced 4-bit value to result.

### *C. Triple Data Encryption Standard*

Triple Data Encryption Standard (Triple-DES) [4] was constructed by IBM. This algorithm is 3 times slow than DES. The data encryption standard is broken with brute force attacks, so the developers is used DES 3 times to strength the algorithm.

The work plan of algorithm is same with DES but key size is extended to 128-bit. 128-bit key is divided to 64-bit, first 64-bit key is used in the first and third DES, second 64-bit key is use in the second DES (Figure 5).

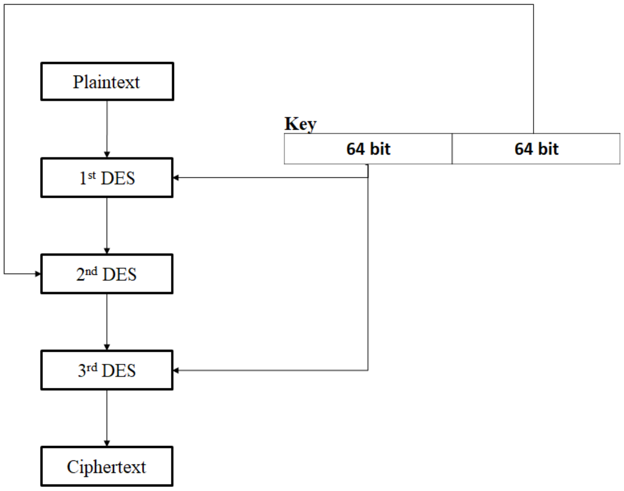

Figure 5: Triple Data Encryption Standard encryption schema

### *D. Advanced Encryption Standard Algorithm*

Joan Daeman and Vincent Rijmen joined the competition in 1997, which was announced by the National Institute of Standards and Technology. They won the competition with Rijndael algorithm, the name of the algorithm is generated from their name. Rijndael algorithm have been advanced encryption standard instead of data encryption standard since 2000. New algorithm has different size of key, is fast and more robust against attack, can be used on variety of application both software and hardware.

Advanced Encryption Standard (AES) is a block cipher algorithm, encrypts the 128-bits data blocks with 128-bits, 192-bits or 256-bits key. It has different round number according to key size. The round number for 128-bits, 192-bits and 256-bits key is 10, 12, and 14 respectively.

Each round of Advanced Encryption Standard has four main steps: Sub Bytes, Shift Rows, Mix Columns, Add Round Key.

**Sub Bytes:** Each byte of matrices converts to different byte with substitution table (S-box). This step is non-linear and robust against differential and linear crypto analysis.

**Shift Rows:** Every row except the first row of the matrix is shifted to the left as a cycle using byte with different offsets.

**Mix Columns:** Each column is multiplied by a specific linear transformation function to obtain a new column.

**Add Round Key:** The generated loop key is added to the bitwise exclusive-OR (XOR) operation with the result of the upper step at the end of each turn.

AES algorithm generates cipher text from plain text after sub bytes, shift rows, mix column, add round key operations in sequence in a loop then return to sub byte step for last round as shown in Figure 6. Last round includes sub bytes, shift rows and add round key step.

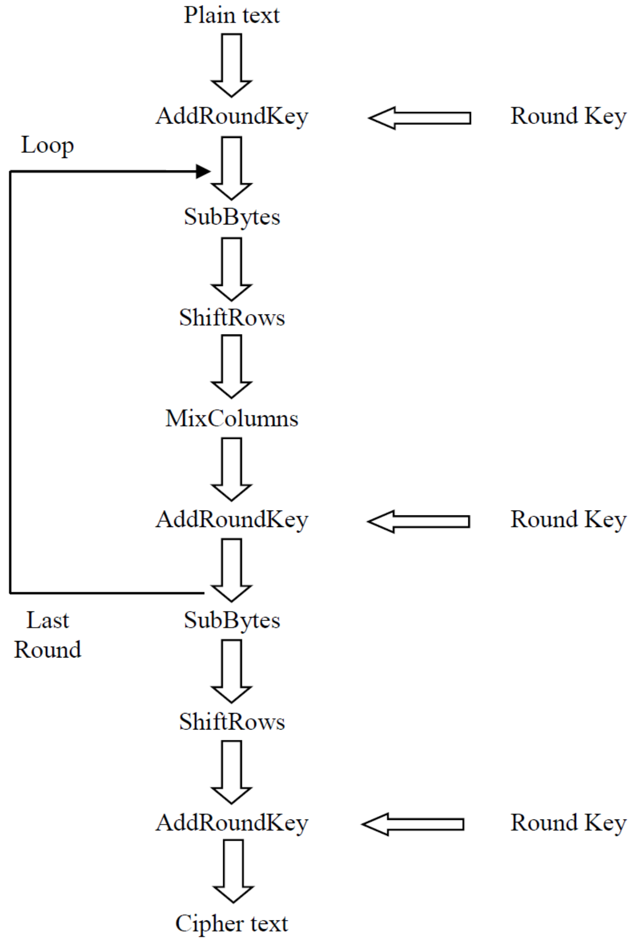

Figure 6: Advanced Encryption Standard encryption schema

## IV. Comparison Of Cryptography Algorithms

Data Encryption Standard (DES), Triple Data Encryption Standard (3-DES), Advanced Encryption Standard (AES) that are symmetric block encryption algorithms and Rivest-Shamir-Adleman (RSA) is asymmetric encryption are compared according to ten factors that are basic, principle, plaintext size, key size, loop number, security, speed, power consumption, hardware or software application, crypto analysis (Table I).

According to aim of the application, one of the cryptography algorithms become the best to use in application. In mobile payment systems, important factors are speed, power consumption, security and crypto analysis. The consumers prefer to use mobile payment system because of speed. Therefore, the most important factor is speed and when we look speed performance of the Triple-DES and RSA are more slowly than the other algorithms. When we look at power consumption of the algorithms, DES and AES has minimum consumption. Security option of the algorithms is related with the key size, AES has different option on key size. Therefore, AES seems to be more robust against attacks. In addition, AES is more robust truncated differential, interpolation, and square attacks.

As shown in Table 1, AES is more robust to attacks and used both hardware and software applications with minimum consumption, maximum speed.

Table 1: Comparison of Algorithms [6-9]

| **Factor** | **Cryptography Algorithms** | | | |
|---|---|---|---|---|
| | **DES** | **Triple-DES** | **AES** | **RSA** |
| Basic | Data block is separate in two blocks. | Data block is separate in two blocks. | Data block is used in one block. | Two different keys |
| Principle | Feistel encryption | Feistel encryption | Substitution and permutation | It is difficult to deal with big integer. |
| Plaintext | 64-bit | 64-bit | 128-bit, 192-bit, 256-bit | Minimum 512-bit |
| Key size | 56-bit | 112-bit | 128-bit, 192-bit, 256-bit | >=1024-bit |
| Loop number | 16 | 48 | 10, 12, 14 | - |
| Security | Key size is not enough for security | More secure than DES | Based on key size | Based on big prime number |
| Speed | Slow | More slowly | Fast | More slowly |
| Power consumption | Minimum | Maximum | Minimum | Maximum |
| Hardware or Software application | Hardware | Hardware | Both | Not efficient |
| Crypto analysis | Weak against differential and linear crypto analysis, weak substitution table | Weak against differential analysis, plaintext can find with brute-force and differential crypto attacks | Robust truncated differential, interpolation, square attacks | Weak against brute force and oracle attacks |

Table 2: Time consumption of Algorithms

| **Cryptography Algorithm** | **Factor: Time consumption** |
|---|---|
| | **Execution Time** |
| DES | 3 second |
| Triple-DES | ~ 9 second |
| RSA | 7 second |
| AES-128 | 791 milliseconds |
| AES-192 | 830 milliseconds |
| AES-256 | 845 milliseconds |

As shown in Table 2, we compare time consumption of all cryptography algorithms in c programming language on NetBeans IDE 8 platform. Same data is used as input in all algorithms to encrypt. We get different output values (ciphertext) in different time. As a result of Table II, symmetric cryptography algorithms have better performance on speed than asymmetric cryptography algorithms. As shown in Table II, the best cryptography algorithm is Advanced Encryption Standard according to execution time of algorithm during encryption.

It is understood that, AES is most suitable algorithm for mobile application according to speed criteria. Therefore, we generated two different scenarios to improve the security side of AES. In both scenario, two table are generated for key and data. The values in each table are generated randomly between 0 (0x00) and 256 (0xFF). These table are generated once time in application and it takes approximately 609 milliseconds.

**Algorithm #1: encrypt**

```
procedure main()
    state <- GenerateHex(plaintext)
    plaintext <- CheckSize(state)
    ciphertext <- enryptwithAES(plaintext)
    ciphertext <- EncryptPartial(ciphertext)
    return 0
end procedure
```

**Algorithm #1: decrypt**

```
procedure main()
    ciphertext <- DecryptPartial(ciphertext)
    state <- decryptwithAES(ciphertext)
    state <- CheckSize(state)
    plaintext <- SolveHex(state)
    return 0
end procedure
```

In first algorithm which is given above, ciphertext was generated from AES encryption. Generated ciphertext is divided in to blocks. A bitwise logical exclusive-or operation is done between two encrypted blocks sequentially except first block. Generated value is as ciphertext of Algorithm #1. The plain text is obtained so that the same operations will be in the reverse order as shown decryption of the Algorithm #1.

**Algorithm #2: encrypt**

```
procedure main()
    hashData <- GetHash()
    state <- GenerateHex(plaintext)
    plaintext <- CheckSize(state)
    ciphertext <- enryptwithAES(plaintext)
    ciphertext <- EncryptPartial(ciphertext)
    text <- AddHash(hashData, ciphertext)
    ciphertext <- GenerateHex(text)
    return 0
end procedure
```

**Algorithm #2: decrypt**

```
procedure main()
    ciphertext <- SolveHex(ciphertext)
    data <- RemoveHash(ciphertext)
    ciphertext <- DecryptPartial(data)
    state <- decryptwithAES(ciphertext)
    state <- CheckSize(state)
    plaintext <- SolveHex(state)
    return 0
end procedure
```

In second proposed algorithm which is given above, use same steps of Algorithm #1. Last version of ciphertext is given in operation with hash data that is generated during execution of algorithm.

We compare two proposed algorithms according to time and storage consumption as shown in Table III. Both scenarios are used same tables to encrypt and decrypt data. When we look both scenarios, storage data capacity is same. Data table has 512 bytes for encrypt, 512 bytes for decrypt for one block. System has 32 blocks so totally 32768 bytes is consumed for data table. Key table also storage with same capacity. All tables are storage as encrypted on system and 65536 bytes (Table III). The second important criteria is time consumption of the algorithm. Although we add new operations to proposed algorithms, time consumption of two proposed algorithms did not exceed the threshold of the mobile applications (Table 3).

Table 3: Comparison of proposed algorithms

| Factor | | Cryptography Algorithms | |
|---|---|---|---|
| | | Algorithm #1 | Algorithm #2 |
| Time consumption | Encryption | ~15 milliseconds | ~17 milliseconds |
| | Decryption | ~16 milliseconds | ~31 milliseconds |
| Storage consumption | | ~ 64 bytes | ~ 64 bytes |

## V. Conclusion

In mobile payment systems, it is essential to store and transmit data in a secure manner, as well as to ensure that these operations are fast. To guarantee the security means to develop new extra operation on security algorithm. Adding extra mathematically operation in system needs more time to finish transaction. Any scenario that exceeds the time limit is not preferred, although it is safe.

In this study, we compared both symmetric and asymmetric cryptography algorithm. Advanced Encryption Standard is the best cryptography algorithm in both security and speed. We proposed two new algorithms to improve security part of the AES with adding new operation. As a result of two scenario, security has been increased and there is no situation that prevents the system to perform quickly.

## Acknowledgment

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 in Science and Engineering, Istanbul University, Istanbul, Turkey.

## REFERENCES

[1] (2013, 31 Mart). Türkiye'deki Ödeme Sistemlerinin Kırılımı: Alternatif Ödeme Sistemleri ve Detayları. http://www.odemesistemleri.org/

[2] RSA Laboratories, PKCS#1 v2.1: RSA Cryptography Standard, EMC Corporation, October 27, 2012.

[3] FIPS 46-3, Data Encryption Standard, Federal Information Processing Standard (FIPS), Publication 46-3, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., October 25, 1999.

[4] FIPS 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Federal Information Processing Standard (FIPS), Special Publication 800-67, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., November, 2017.

[5] FIPS 197, Advanced Encryption Standard, Federal Information Processing Standard (FIPS), Publication 197, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., November 26, 2001.

[6] Mathur M., Kesarwani A., "Comparison Between DES, 3DES, RC2, RC6, Blowfish and AES", Proceedings of National Conference on New Horizons in IT - NCNHIT 2013, 143-148.

[7] Mahajan P., Sachdeva A., "A Study of Encryption Algorithms AES, DES and RSA for Security", Global Journal of Computer Science and Technology Network, Web & Security, Volume 13, Issue 15, Version 1.0, 2013, Online ISSN: 0975-4172,Print ISSN: 0975-4350.

[8] Padmavathi B., Kumari S. R., "A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique", International Journal of Science and Research (IJSR), Volume 2, Issue 4, April 2013, India Online ISSN: 2319-7064.

[9] Singhal S., Singhal N., "A Comparative Analysis of AES and RSA Algorithms", International Journal of Scientific & Engineering Research, Volume 7, Issue 5, May-2016 149, ISSN 2229-5518.