International Journal of Research in Advent Technology, Vol.2, No.11, November2014
E-ISSN: 2321-9637

A Comparative Analysis of Different Encryption Techniques Used In Mobile Cloud Computing

Dr. M Newlin Rajkumar[1], Muhamed Sabir J[2], Dr. V Venkatesa Kumar[3], Bharathi S P[4]
Assistant Professor[1], PG Scholar[2], Assistant Professor [3],PG Scholar[4]
Department of Computer Science and Engineering, Anna University Regional Centre, Coimbatore[1, 2, 3,4]
Email: newlin_rajkumar@yahoo.co.in[1], muhamed.sabir.j@gmail.com[2]

Abstract- In Cloud Computing, a network of remote servers hosted on the internet is used to store, manage, and process data rather than a local server or a personal computer. This helps to reduce network infrastructure and maintenance costs and increases flexibility. It also helps to streamline processes, i.e. get more work done in less time with less number of people. Due to the large increase in the number of mobile users worldwide the requirements for cloud computing on mobile devices also became really important. This gave birth to the idea of Mobile Cloud Computing. Mobile Cloud Computing (MCC) is the combination of cloud computing, wireless networks and mobile computing based on the pay-as-you-use principle. In MCC, data storage and processing are done in the cloud away from the device. Because of this, data security becomes a really important issue. Therefore, for securing data in the cloud various encryption algorithms are used. In this paper, we are presenting a comparative analysis of the features and performances of different encryption techniques used in MCC viz., DES, AES, Blowfish and RSA.

Index Terms- data security; encryption; mobile cloud computing; DES; AES; Blowfish; RSA

1. INTRODUCTION

Cloud computing refers to computing based on the internet in which huge number of remote servers are networked together to allow the access of services and resources online and also for centralized storage of data. Clouds are classified into three public, private and hybrid. In Cloud Computing, users only utilize what they require and only pay for what they use. It mainly focuses on maximizing the effectiveness of shared resources. Cloud resources are not only used by multiple users but also dynamically relocated on demand. It helps IT companies reduce upfront infrastructure cost and provides more flexibility. It also helps enterprises to get their applications up-time to get reduced with less maintenance and improved manageability [1]. Cloud Service Providers uses the use-as-you-go model, i.e. users have to pay for only what they use. NIST defines cloud computing as "a model for enabling ubiquitous and convenient access of the network when needed to a shared group of computing resources like networks, servers, etc. which can be configured that can be quickly setup and released with minimum effort for management and interaction with service providers" [2].

Resources like battery life, storage and bandwidth are very limited for mobile devices. Here comes the application of cloud computing. It allows the users to use the resources provided by cloud service providers such as infrastructure, software and platforms at low cost in a pay-as-you-use fashion. This fact and the increasing popularity of mobiles gave birth to the idea of Mobile Cloud Computing (MCC). It is the combination of cloud computing, mobile computing and wireless networks to bring rich computational resources to network operators, mobile users and cloud computing providers [3]. Its main purpose is to make rich computational resources easily available to mobile users, network operators, and cloud computing service providers. Mobile gadgets like Smart phones, tablets and the concept of cloud computing are merging in the swiftly growing area of mobile cloud computing. According to a recent survey, 1 trillion cloud-ready mobile devices will be available all over the world in less than 4 years. MCC provides mobile users with the facilities of storing and processing data in the clouds, thereby eliminating the need for having a powerful device configuration. This is because of the fact that all computing jobs which are consuming these precious resources can be transferred to the cloud. On the basis of the data accumulated by a recent survey by ABI Research, more than 240 million enterprises will start using cloud services via mobile devices by 2015 [4].

In Mobile Cloud Computing, the data storage and processing are done in the cloud. Therefore the security of the data in the cloud becomes a real issue. This is because of the fact that users may be storing personal and confidential details in the cloud. If the security of the data in the cloud is not strong the data may be compromised. Hence the properties of data integrity and confidentiality are absolutely necessary. For ensuring this, different encryption techniques are employed.

2. DATA ENCRYPTION STANDARD (DES)

DES is an archetypal block cipher - i.e., it takes a string of plaintext bits of fixed-length and converts it via a group of complicated transformations into another cipher text bit string of the same length. It was selected by NIST (National Institute of Standards and Technology) as a Federal Information Processing Standard 46 (FIPS PUB 46). In DES, data are encrypted using a key of 56-bit length. Data is divided into 64-bit blocks. The key consists of 64 bits in total; but, only 56 bits are actually used by the algorithm. 8 bits are used only for checking parity, and are afterwards discarded. Therefore the effective key length is 56 bits.

First the plaintext passes through an initial permutation (IP) phase, which rearranges the bits to a permuted input. It then passes through a phase of 16 rounds of a same function called the Feistel Function which consists of both permutation and substitution functions. The output of the final round (16th) is of 64 bit length. It is a function of the key which is used for encryption and the initial plain text. The right and left halves of this output are interchanged to produce the pre-output. It is then passed through a permutation (FP) which is the inverse of the IP function to produce the Cipher Text

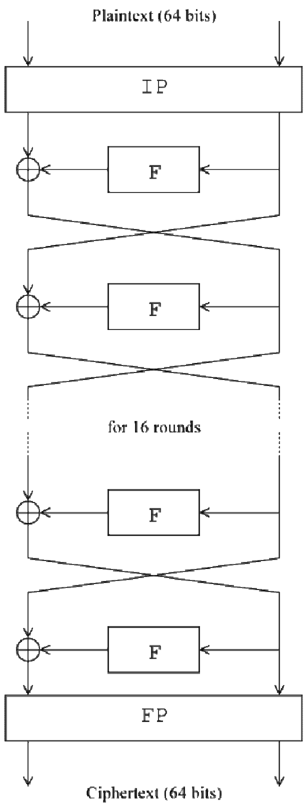

Fig. 1 Overall Feistel Structure of DES

2.1 Feistel Function

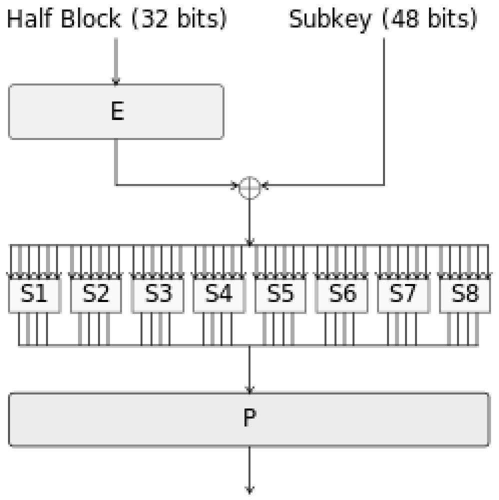

Fig 2. Feistel function of DES

The Feistel function operates on 32 bit blocks at a time and involves four phases:

1. Expansion — using the expansion permutation (E) the 32-bit block is expanded to 48 bits by duplicating 16 bits which is half of the bits. The output will contain eight pieces each of length 6 bits, containing a copy of 4 bits from corresponding input, and a copy of the very next bit from each of the input bits to both sides.

2. Key mixing — the result is XORed with a subkey. 16 48-bit subkeys — one for each round — are derived from the main key using the a key schedule

3. Substitution — after the mixing is done in the subkey, the block is split into eight pieces of 6 bit length before feeding them to transformation boxes called S-boxes or substitution boxes. Each of these eight S-boxes substitutes its input bits of length 6 bits with four output bits based on a lookup table. The main component of the security provided by DES is the S-boxes. Without S-boxes the cipher text will be easily crackable.

4. Permutations — finally, all of the 32 outputs of the S-boxes are transformed based on a fixed permutation, called the P-box. This is designed in such a way that, after this permutation step, each output bit from the substitution boxes are distributed over the 4 different S boxes in the next round.

3. AES (ADVANCED ENCRYPTION STANDARD)

The Advanced Encryption Standard (AES) is a symmetric block cipher algorithm. It was accepted by the U.S. government as a standard for encryption and decryption of classified data. It was developed by Joan Daemen and Vincent Rijmen, two cryptographers from Belgium. NIST started a five-year algorithm development program to substitute the DES and Triple DES in 1997. The NIST selected 15 algorithms and closely reviewed them. Finally, after thorough evaluation, the Rijndael design was selected. In 2001, NIST approved the AES as FIPS PUB 197, which specifies application of the algorithm to all sensitive and classified data [5].

AES is based on substitution-permutation network (SPN), a design principle which is a combination of both permutation and substitution, and is swifter both in software and hardware. It has blocks of size 128 bits and keys of size 128,192 and 256 bits. The maximum allowed size of the key is unlimited, but the maximum block size is limited to 256 bits. It does not use a Feistel Network like the DES standard. AES operates on a 4×4 column-major order (CMO) matrix of bytes, called the *state*. Some versions of Rijndael have blocks of larger size and have additional columns in the state matrix [6].

The key size used in an AES cipher denotes the number of times transformation rounds that convert the plaintext, into the cipher text are repeated. The numbers of cycles of repetition are:

- 10 cycles for 128-bit keys.
- 12 cycles for 192-bit keys.
- 14 cycles for 256-bit keys.

Each round consists of four different but similar steps, including a step which depends on the encryption key. Using the same key the same steps are applied in reverse to decrypt the cipher text back into the original plaintext.
The four steps in each of the rounds are:

1. Key Expansion— Keys for each round are derived from the cipher key using key schedule.
2. AddRoundKey— Using bitwise XOR each byte of the state is clubbed with a block of the round key. This is called the Initial Round
3. Rounds
   i. SubBytes— here according to a lookup table each byte is substituted with another one.
   ii. ShiftRows— final three rows of the state matrix are shifted cyclically a particular number of steps.
   iii. MixColumns— Combines the four bytes in each column.
   iv. AddRoundKey
4. Final Round
   1. SubBytes
   2. ShiftRows
   3. AddRoundKey.

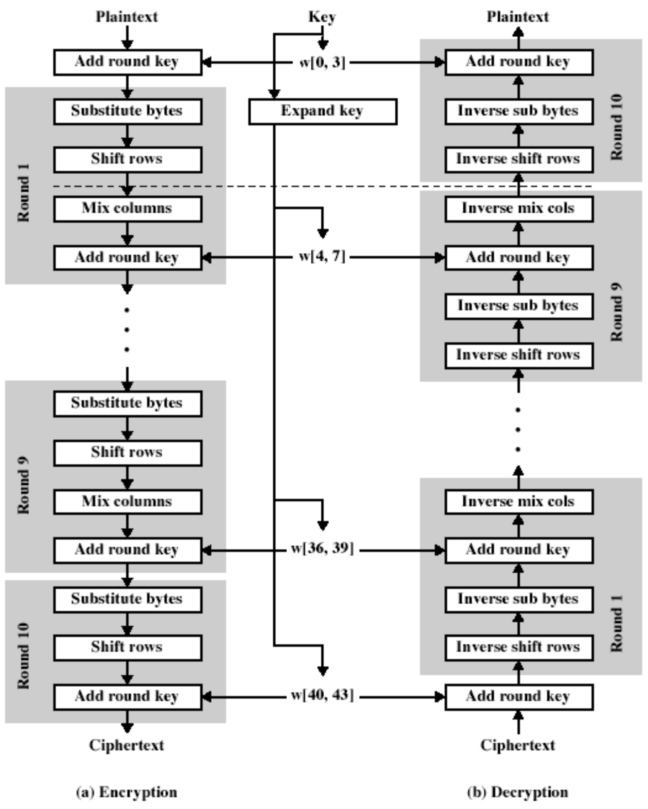

Fig 3. Overall Structure of AES

4. RSA (RIVEST-SHAMIR-ADLEMAN) CRYPTO SYSTEM

RSA cryptosystem is named after its developers Ron Rivest, Adi Shamir and Leonard Adleman, who described the algorithm first in 1977. It is a public key cryptosystem in which the encryption key is public and differs from a secretly kept decryption key called the private key. This asymmetry is based on the problem in the practicality of factoring the product of two large prime numbers.

The RSA algorithm consists of three steps: key generation, encryption and decryption.

4.1 Key generation

RSA involves a *public key* (a key known to everyone) and a *private key*. The public key is used for encrypting messages and it can be decrypted only using the private key in a particular amount of time. The keys are generated using the following steps:

1. First, we should choose two distinct prime numbers p and q at random. The numbers should be of similar bit length.
2. Calculate the value of n using the equation, $n = pq$. The value of n is used as the modulus for both the public and private keys and its length is the key length.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function.
4. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are coprime. e is released as the public key exponent. If the length of e is short the encryption will be more efficient. However, much smaller value of e (such as 3) have been shown to be less secure.
5. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$; i.e., d is the multiplicative inverse of e (modulo $\varphi(n)$).
   - This can be clearly stated as: solve for d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$
   - d is kept as the private key exponent.

The *public key* consists of the modulus n and the public (or encryption) exponent e. The *private key* consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and $\varphi(n)$ must also be kept secret because they can be used to calculate d.

4.2 Encryption

Alice transmits her public key (n, e) to Bob and keeps the private key d secret.

If Bob wants to send a message M to Alice, he does the following.

He first turns M into an integer m, such that $0 \le m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c \equiv m^e \pmod{n}$$

This can be done efficiently, even for 500-bit numbers, using Modular exponentiation. Bob then transmits c to Alice.

4.3 Decryption

Alice can recover m from c by using her private key exponent d by calculating

$$m \equiv c^d \pmod{n}$$

Given m, she can recover the original message M by reversing the padding scheme

5. BLOWFISH

Blowfish is a symmetric-key block cipher, designed by Bruce Schneier in 1993. It provides good encryption rate in the software and no effective cryptanalysis has been found until now [7].It was originally designed as a general-purpose algorithm, to replace the DES and was free of the problems and constraints other algorithms had at that time. When Blowfish was released, many other algorithms were proprietary, or were commercial or government secrets. But Schneier placed blowfish in the public domain and stated that it is unpatented and can be freely used by anyone. The main features of Blowfish are key-dependent S-boxes and the highly complex key schedule. It has a 64-bit block size and a variable key length from 32 bits up to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes.

5.1 Encryption

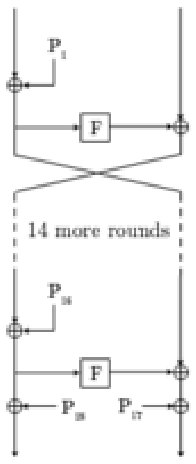

Fig 4. Feistel Structure of Blowfish

Each line in the Fig.4 represents 32 bits. The algorithm maintains two subkey arrays: the P-array having 18 entries and four S-boxes having 256 entries each. The S-boxes takes inputs of size 8-bits and produces 32-bit outputs. In every round, one entry from the P-array is used, and after the last round, one of the two remaining P-entries which are un-used is XORed with each half of the data block.

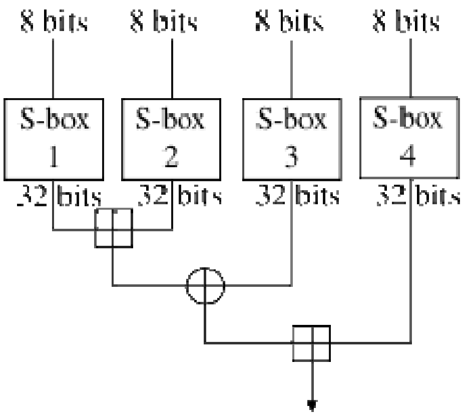

Fig 5. Feistel funcion of Blowfish

The 32-bit input is broken into four eight-bit quarters by the F-function. It then feeds the quarters thus obtained to the S-boxes. The outputs then will be added modulo 2^{32} and finally fed to an XOR to produce the final output, which is of 32-bit length.

5.2 Decryption

Decryption is just like the encryption, except for the fact that the order in which P1, P2,..., P18 are used is reversed. This is not so evident since xor is both associative and commutative. A common mistake is to use reverse order of encryption as decryption algorithm.

Blowfish is found to be susceptible to attacks on reflectively weak keys. This means users must be careful in selecting keys as there is a class of keys known to be weak, or switch to more modern replacements like the Advanced Encryption Standard, or Blowfish's more modern successors viz. twofish and threefish.

6 PERFORMANCE ANALYSIS

According to[8], Blowfish algorithm has better performance in comparison with AES and DES. Simulation results presented in the same paper also suggests that it has better throughput, takes less processing time and consumes less battery power.

In contrast to this while encrypting and decrypting video files AES is found to have better throughput and performance according to the results obtained in [9].

Based on experiments conducted on text files by [10] we can understand the fact that DES takes less time for encryption, whereas AES has least memory usage. We can also observe that RSA takes the longest for encryption and consumes most memory also. The main factors considered in this study were encryption time, memory usage and output byte.

[11] analyses the performance of DES and RSA. Out of these two DES is found to have better throughput in comparison with RSA.

7. FEATURES COMPARISON

Algorithm	*DES*	*AES*	*Blowfish*	*RSA*
Developed by	IBM (1977)	Jon Daemen & Vincent Rijmen (2000)	Bruce Schneier (1993)	Rivest-Shamir-Adleman (1977)
Key Size (in bits)	56	128,192,256	32-448	1024-4096
Block size (in bits)	64	128	64	Variable
Algorithm Structure	Feistel network	Substitution Permutation network	Feistel network	-
No. of Rounds	16	10,12,14	16	1
Vulnerable to	Brute force attack, Differential Cryptanalysis, Linear Cryptanalysis	Side Channel Attacks	Second order differential attacks	Timing attacks, Adaptive chosen cipher text attacks

8. CONCLUSION

Mobile Cloud Computing is an emerging area in Cloud Computing which is gaining huge popularity worldwide and in the coming years it is only going to increase. The main threat in MCC is the security of data stored in the cloud. We have done a detailed study about three of the most popular symmetric cryptographic algorithms DSA, AES, Blowfish and about the asymmetric RSA algorithm in this paper. A comparative analysis of features is given in Table 1. Every encryption method is unique in its own away and has its own advantages and disadvantages. From the performance analysis presented in the paper, we can understand that some techniques are best suited for video files where as some for texts. Since we are dealing with mobile devices our environment is highly resource constraint. Hence, we have to give attention on the power consumed also because battery power is really critical in mobile devices. If these techniques are combined properly the security can be further enhanced. Future work can be conducted in this regard.

ACKNOWLEDGEMENT

I express my sincere gratitude to my guide Dr. M Newlin Rajkumar and my co-guide Dr.V Venkatesa Kumar for their guidance and support in completing this paper.

REFERENCES

[1] http://aws.amazon.com/what-is-cloud-computing/ Retrieved 2014-10-28

[2] http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf Retrieved 24 July 2011

[3] Abolfazli et.al, "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges," *IEEE Communications Surveys & Tutorials*, (pp): 1–32, July 2013.

[4]http://www.crn.com/news/mobility/222300633/report-mobile-cloud-computing-a-5-billion-opportunity.htm

[5]http://www.techopedia.com/definition/1763/advanced-encryption-standard-aes

[6] Bruce Schneier et.al, "The Twofish Team's Final Comments on AES Selection", May 2000

[7] Bruce Schneier , "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", *Cambridge Security Workshop Proceedings* (Springer-Verlag),pp.191–204,1993.

[8] P. C. Mandal, "Evaluation of performance of the Symmetric Key Algorithms: DES, 3DES, AES and Blowfish," *J. Glob. Res. Comput. Sci.*, vol. 3, no. 8, pp. 67–70, 2012.

[9] S. Pavithra, "Performance Evaluation of Symmetric Algorithms," *J. Glob. Res. Comput. Sci.*, vol. 3, no. 8, pp. 43–45, 2012.

[10] S. M. Seth and R. Mishra, "Comparative Analysis of DES, AES, RSA Encryption Algorithms," *IJCST*, vol. 2, no. 2, pp. 292–294, Jun. 2011.

[11] A. Kumar and B. Bahal, "Comparative Analysis," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 7, pp. 386–391, Jul. 2012.