International Journal of Computing Algorithm
Volume: 05 Issue: 01 June 2016, Page No. 29- 33
ISSN: 2278-2397

# Evolution of Cryptographic Algorithms and Performance Parameters

C S Selin Chandra[1], S. Sujin Lal[2], A. Saranya[1]
[1]Research scholar PG & Research Dept of Computer science, D.B.Jain College
[2]Assistant Professor PG & Research Dept of Computer science, D.B.Jain College
Email: selin.chandra2@gmail.com, lalsujin@gmail.com, saran91aji@gmail.com

Abstract - Internet and networks applications are growing very fast. Most of confidential data is circulated through networks as electronic data for achieving faster communication. In data communication Information Security became a major problem. The information is sensitive part of the organization. Any loss or threat to information can prove to be great loss to the organization. Encryption technique plays an inevitable role in information security systems. Cryptographic ciphers have an important role for providing security to these confidential data against unauthorized attacks. Though security is an important factor, there are various factors that can affect the performance and selection of cryptographic algorithms during the practical implementation of these cryptographic ciphers for various applications. This paper discuss about the data and procedures for selecting cryptographic algorithms with respect to performance.

Keywords - Cryptography, Security, Authentication, Performance

## I. INTRODUCTION

Cryptography is a powerful tool used for the network security. Cryptography algorithms play an important role in information security. Cryptology has been incorporated into smart cards for financial dealings, operating systems, web browsing, mobile phones and electronic identity cards. Cryptography can be divided into Symmetric and Asymmetric key cryptography. In Symmetric key encryption only one key is used to encrypt and decrypt data. Symmetric algorithms are also called as secret key algorithms, conventional algorithms, shared algorithms, private key algorithms or one key algorithm. In Asymmetric key encryption, two keys are used; private keys and public keys. They are related to each other. One key is used for encryption and a different but related key is used for decryption. Public key is known to the public and private key is known only to the user. Symmetric algorithms are of two types: Block Ciphers and Stream Ciphers. The block ciphers are operating on data in groups or blocks. Here the size of the block is of fixed size for encryption. Stream ciphers are operating on a single bit at a time. Here continuous stream is passed for encryption and decryption. Since symmetric encryption requires less computational processing power, they are near to 1000 times faster than asymmetric techniques.

## II. VARIOUS CRYPTOGRAPHIC ALGORITHMS

Symmetric cryptographic algorithms (Private Key systems)
The private key systems in common use today are,

*1) ROT13*

ROT13 is a simple cryptography algorithm which has no key, and it is not secure. [1]

*2) Crypt*

The original UNIX encryption program which is modeled on the German Enigma encryption machines. Crypt uses a variable-length key. Some programs can automatically decrypt crypt-encrypted files without prior knowledge of the key or the plaintext. Crypt is not secure. [1]

*3) DES*

The Data Encryption Standard (DES) an encryption algorithm developed in the 1970s by the National Bureau of Standards and Technology (since renamed the National Institute of Standards and Technology) and IBM [1]. It was the first encryption standard published by NIST [2]. It is based on the IBM proposed algorithm called Lucifer. It uses one 64-bit key. Out of 64 bits, 56 bits make up the independent key, which determine the exact cryptography transformation; 8 bits are used for error detection. The main operations are bit permutations and substitution in one round of DES. Six different permutation operations are used both in key expansion part and cipher part. Decryption of DES algorithm is similar to encryption, only the round keys are applied in reverse order. The output is a 64-bit block of cipher text [2]. DES is based on two fundamental attributes of cryptography: Substitution (confusion) and transposition (Diffusion). DES consists of 16 steps, each of which called as a Round [5].
Different forms of DES algorithm

*a) DESX*

DESX is a stronger variation of the DES encryption algorithm. In DESX, the input plaintext is bitwise XORed with 64 bits of additional key material before encryption with DES and the output is also bitwise XORed with another 64 bits of key material [4].

*b) DOUBLE DES*

It is also called 2DES. Its process is the same as DES but repeated the same process 2 times using two keys K1 and K2. First it takes plain text, produced the cipher text using K1 and then take up the cipher text as input, produced another cipher text using K2 [5].

*c) 3DES*

As an enhancement of DES, the3DES (Triple DES) encryption standard was proposed. It uses 64 bit block size with 192 bits of key size. 3DES encryption method and DES encryption method are similar, but applied 3 times for increasing the encryption level and the average safe time[6].However 3DES is slower than other block cipher methods [2].

It comes in two flavors: One that uses three keys and another that uses two keys. The plain text block P is first encrypted with a key K1, then encrypted with second key K2, and finally with third key K3, where K1, K2 and K3 are different from

International Journal of Computing Algorithm
Volume: 05 Issue: 01 June 2016, Page No. 29- 33
ISSN: 2278-2397

each other. To decrypt the cipher text C and obtain the plain text, we need to perform the operation P= DK3 (DK2 (DK1(C))). But in Triple DES with two keys the algorithms work as follows:

*Step 1:* Encrypt the plain text with key K1. Thus, we have EK1(p) .
*Step 2:* Decrypt the output of step1 above with key K2.Thus, we have DK2 (EK1 (P)).
*Step 3:* Finally, encrypt the output of step 2 again with a key K1. Thus, we have EK1 (DK2 (EK1 (P))) [5].

*d) RC2*

A block cipher originally developed by Ronald Rivest and kept as a trade secret by RSA Data Security. This algorithm was revealed by an anonymous Usenet posting in 1996 and appears to be reasonably strong (although there are some particular keys that are weak). RC2 is sold with an implementation that allows keys between 1 and 2048 bits. The RC2mail key length is often limited to 40 bits in software that is sold for export. Unfortunately, a 40-bit key is vulnerable to a brute force attack [1].

*e) RC4*

A stream cipher originally developed by Ronald Rivest and kept as a trade secret by RSA Data Security. This algorithm was revealed by an anonymous Usenet posting in 1994 and appears to be reasonably strong (although there are some particular keys that are weak). RC4 is sold with an implementation that allows keys between 1 and 2048 bits. The RC4 key length is often limited to 40 bits in software that is sold for export. Unfortunately, a 40-bit key is vulnerable to a brute force attack [1].

*f) RC5*

A block cipher developed by Ronald Rivest and published in 1994. RC5 allows a user-defined key length, data block size, and number of encryption rounds [1].

*g) IDEA*

The International Data Encryption Algorithm (IDEA) developed in Zurich, Switzerland by James L. Massey and Xuejia Lai and published in 1990. IDEA uses a 128-bit key. IDEA is used by the popular program PGP to encrypt files and electronic mail. [1].The algorithm consists of nine phases: eight identical phases and a final transformation phase. IDEA is supposed to have very good cryptanalytic properties, there by combining efficiency with acceptable security [10].

h) *Skipjack*

A classified (SECRET) algorithm developed by the National Security Agency (NSA). Reportedly, a Top Secret security clearance is required to see the algorithm's source code and design specifications. Skipjack is the algorithm used by the Clipper encryption chip. It uses an 80-bit key [1].

i) *AES*

Advanced Encryption Standard is the new encryption standard recommended by NIST to replace DES [2][4]. Advanced Encryption Standard (AES) also known as the Rijndael algorithm is a symmetric block cipher. Rijndael algorithm was selected in 1997 after a competition to select the best encryption standard. Brute force attack is the only effective attack known against it, in which the attacker tries to test all the characters combinations to unlock the encryption [11].

It has variable key length of 128, 192, or 256 bits; default 256. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES has been carefully tested for many security applications [2]. It is approved by US government in 2000 for encryption of sensitive but unclassified data. It is well suited for implementation in hardware and software [4].

j) *Blowfish*:

It is one of the most public domain encryption algorithms [2]. Blowfish was designed in 1993 by Bruce Schneider as a fast alternative to existing encryption algorithms. Blowfish is a symmetric key block cipher that uses a 64 bit block size and variable key length. It takes a variable-length key from 32 bits to 448 bits.

Blowfish has variants of 14 rounds or less. Blowfish is a very secure cipher but it has been replaced by Twofish and Rijndael due to its small 64 bit block size. Blowfish is one of the fastest block ciphers which has developed to date. Blowfish was created to allow anyone to use encryption free of patents and copyrights. Blowfish has remained in the public domain to this day. No attack is known to be successful against it, though it suffers from weak keys problem [2].

*k) Two fish:*

The Two fish encryption algorithm was designed in order to make the Advanced Encryption Standard (AES). Two fish is a symmetric block code which employs an identical key for encryption and decryption of data. The block size of a Two fish algorithm is 128 bits, and allows a key of length up to 256 bits [7].

*l) UMARAM:*

The UMARAM was designed by Ramesh G and R.Umarani in the year 2010 [6]. To encrypt a plaintext of 512-bits, this algorithm uses a key size of 512-bits during the 16-rounds. Depending on S-BOX, different shift processes, XOR-Gate, and AND-Gate, a series of transformations have been used in this algorithm. The S-Box is used to map the input code to another code at the output. It is a matrix of 16 X 16 X 16. [6].

*m) UR5:*

UR5 algorithm was designed by G.Ramesh and Dr. R. Umarani in the end of the year 2010. A block encryption algorithm is proposed in this approach. Depending on S-BOX, XOR Gate, and AND Gate, a series of transformations have been used in this Algorithm. The UR5 algorithm encrypts a plaintext of size 64-bits by using a key size of 64-bits. It consists of eight rounds for encryption or decryption process. It overcomes some drawbacks of the other algorithms. Because of not using the same key with other packets within a message, this algorithm is more efficient and useable for the Wireless Local Area Network. The algorithm is simple and helps to avoid the hackers. The backbone of this algorithm is S-BOX generation. It has eight columns and 256 rows; each element consists of 8-bits. It replaces the input by another code to the output [8].

International Journal of Computing Algorithm
Volume: 05 Issue: 01 June 2016, Page No. 29- 33
ISSN: 2278-2397

n) *Camellia***:**
Camellia was jointly developed by Nippon Telegraph and Telephone Corporation and Mitsubishi electronics corporation algorithm [8]. This algorithm specifies the 128-bit block size and 128, 192 and 256 bit key sizes. It was based on feistel network cipher with 18 or 24 rounds.

**Asymmetric cryptographic algorithms (Public key systems):**

The public key systems in common use today are,

*1) Diffie-Hellman:*
A system for exchanging cryptographic keys between active parties. Diffie-Hellman is not actually a method of encryption and decryption, but a method of developing and exchanging a shared private key over a public communications channel. In effect, the two parties agree to some common numerical values, and then each party creates a key. Mathematical transformations of the keys are exchanged. Each party can then calculate a third session key that cannot easily be derived by an attacker who knows both exchanged values.

Several versions of this protocol exist, involving a differing number of parties and different transformations. Particular care must be exercised in the choice of some of the numbers and calculations used or the exchange can be easily compromised. The Diffie-Hellman algorithm is frequently used as the basis for exchanging cryptographic keys for encrypting a communications link. The key may be any length, depending on the particular implementation used. Longer keys are generally more secure [1].

*2) RSA:*
The well-known public key cryptography system developed by MIT professors Ronald Rivest and Adi Shamir, and by USC professor Leonard Adleman in 1977 [1]. RSA algorithm is the most popular and proven asymmetric key cryptographic algorithm [5]. RSA can be used both for encrypting information and as the basis of a digital signature system. Digital signatures can be used to prove the authorship and authenticity of digital information. The key may be any length, depending on the particular implementation used. Longer keys are generally considered to be more secure [1].

The RSA algorithm is based on the mathematical fact that it is easy to find and multiply large prime numbers together, but it is extremely difficult to factor their product. The private and public keys in the RSA are based on very large prime numbers. However, the real challenge in the case of RSA is the selection and generation of the public and private keys [5].

3) *ElGamal***:**
Another algorithm based on exponentiation and modular arithmetic. ElGamal may be used for encryption and digital signatures in a manner similar to the RSA algorithm. Longer keys are generally considered to be more secure [1].

4) *DSA***:**
The Digital Signature Algorithm developed by NSA and adopted as a Federal Information Processing Standard by NIST. Although the DSA key may be any length, only keys between 512 and 1024 bits are permitted under the FIPS. As specified, DSA can only be used for digital signatures, although it is possible to use DSA implementations for encryption as well. The DSA is sometimes referred to as the DSS, in the same manner as the DEA is usually referred to as the DES [1].

## III. MEASURING THE PARAMETERS AND ANALYSING THE PERFORMANCE OF ALGORITHMS

One of the important components of any encryption algorithm is Performance. This part gives a description about simulation environment, system components, and various metrics for the performance and the procedure for analyzing the performance of algorithms.

*1) Different System parameters:*
Performance of algorithm can be analyzed by,

- Using 2 architectures such as wired architecture and wireless architecture.
- Using different system configuration such as laptop, standalone pc, Networked pc to get better comparison results.
- Using Different operating systems.

*2) Various Metrics For The Performance:*
Evaluation of the performance of proposed algorithm is based on the several various metrics which are best suited for the cryptographic algorithms. Some selected metrics for the evaluation are Encryption time, decryption time, Throughput of encryption, Throughput of decryption, CPU process time, CPU clock cycles Power consumption, Memory Utilization.

The total time taken by an algorithm to produce a cipher text from plain text is known as Encryption time. It is used to calculate the throughput of the encrypted algorithm. It gives the rate of encryption [10]. Decryption time is the time taken by an algorithm to produce plain text from cipher text. It is used to calculate the throughput of the decrypted algorithm. It gives the rate of decryption [10]. The speed of encryption is calculated from throughput of the encryption [10]. Throughput of encryption = Plain Text (MB) / Encryption time [2]. Throughput of the encryption algorithm and the power consumption algorithm are inversely proportional to each other. If there is an increase in the throughput of the encryption algorithm, there is a decrease in the power consumption algorithm [10].

The speed of decryption is calculated from throughput of the decryption [10]. Throughput of decryption = Plain Text (MB) / Encryption time [2]. Throughput of the decryption algorithm and the power consumption algorithm are inversely proportional to each other. If there is an increase in the throughput of the decryption algorithm, there is a decrease in the power consumption algorithm [10]. The CPU process time is the time when the load of the CPU is happened for the particular process of calculations. [9]. The CPU load will be higher if the the encryption process is used more CPU time. [10]. The CPU clock cycles represent the energy consumption of the CPU when encryption operations are going on. Each cycle of CPU will consume a small amount of energy. Power consumption is the total power that required by the encryption

International Journal of Computing Algorithm
Volume: 05 Issue: 01 June 2016, Page No. 29- 33
ISSN: 2278-2397

and decryption algorithm. It was estimated based on the throughput of the encryption and decryption algorithm. When increase in the throughput of the encryption/decryption algorithm, there is a decrease in the power consumption algorithm [10]. Memory Utilization is the analysis of memory requirement for the encryption and decryption [10].
The formula to calculate the average encryption time

$$AverageTime = \frac{1}{Nm}\sum_{i=1}^{Nm}\frac{M}{t}\ (Kb/s)$$

Nm=Number of Messages
M=Message size (Kb)
t=Time taken to Encrypt Message M
Energy consumption for encryption and decryption can be measured in many ways [9] [10]. These methods as follows: The First method used to measure energy consumption is to assume that an average amount of energy is consumed by normal operations and to test the extra energy consumed by an encryption algorithms. This method simply monitors the level of the percentage of remaining battery that can computed by equations
The battery life consumed in percentage for one run = Change_in_Batterylife / No_of_Runs

Average battery Consumed per iteration =

$$\sum_{1}^{N}\frac{\text{Battery\_consumed/Iteration}}{\text{No\_of\_Runs}}$$

The second method of security primitives can also be measured by counting the amount of computing cycles which are used in computations related to cryptographic operations. For computation of the energy cost of encryption, we use the same techniques as described in using the following equations.
Bcost_encryption (ampere-cycle)= τ * I

$$Tenergy\_cost\ (ampere - Seconds) = \frac{B\_cost\_encryption}{F(\frac{Cycles}{Sec})}$$

Ecost(Joule) = Tenergy_cost (ampere-seconds) * V
Where Bcost-encryption: a basic cost of Encryption
(ampere-cycle): The total number of clockcycles.
I: the average current drawn by each CPU clock cycle
Tenergy_cost: The total energy cost(ampere-seconds),
F: Clcok frequency(cycles/sec)
Ecost(joule): the energy cost(consumed)

By using the cycles, the operating voltage of the CPU, and the average current drawn for each cycle, we can calculate the energy consumption of cryptographic functions. So, the amount of energy consumed by program P to achieve its goal (encryption or decryption) is given by

E= VCC * Ip * Nc * τ

Where Nc – The number of clock cycles and VCC – The supply voltage of the system
Ip – the average current in amperes drawn from the power source for T seconds.

*3) Procedure For Analysing Performance:*
Several experimental procedures are used for analyzing performance.

a) Files of same type with different packet size: performance are compared by encrypting input files of varying sizes and their encryption time is calculated [2], for example : Different file sizes ranging from 40 Kb to 8000Kb [10].
b) Files with different Data types [3]: Different data type files like audio, image, textual and video of same file size are chosen and encryption time of different cipher algorithms is calculated for these data types. For all executions of a specific cipher algorithm, varying parameter is data type and constant parameter is key size.
c) Different types of files based on varying key size [4]: Different data type files like audio, image, textual and video of same file size and varying key size are chosen and encryption time of different cipher algorithms is calculated for these data types, for example .exe(Executable file), .doc(document file), .wmv(window media video), avi (audio video interface).
d) File with different data densities [3]: This study is taken to check whether the encryption depends on density of data or not. Encryption rate is evaluated for the two different data density file For a cipher algorithm, key size and block mode are kept at bare minimal parameters, for example a sparse file of 69MB and a dense file of 58.5MB.
e) Encryption Algorithms with different key sizes: Different key sizes are employed to trace the performance of the selected algorithms [10]. This study will analyze the effect of changing the size of encryption key on encryption time, for example BMP file of 50.5MB is taken and different cipher algorithms are executed for different size of keys supported [3].
f) Performance of algorithm in different web browsers: In this study, one can use different Web browsers like Internet Explorer, Mozilla Firefox, Opera, Netscape Navigator and Google Chrome in order to determine which type of algorithm is suitable to which type of Web browser in terms of their performance and compatibility [6].

*4) Implementation:*
The algorithm's simulation program can be implement using in one of the languages such as JAVA [2], C# [7], ASP [6].

## IV. PURPOSE FOR ANALYSING PERFORMANCE OF ALGORITHMS

Encryption makes the modern world go round. Every time we make a mobile phone call, buy something with a credit card in a shop or on the web, or even get cash from an ATM, encryption bestows upon that transaction the confidentiality and security to make it possible. If only the encryption algorithm cryptographically strong, it can be used in secure transaction. Cryptographically strong would seem to mean that the described method has some kind of maturity, perhaps even approved for use against different kinds of systematic attacks in theory and/or practice. Indeed, that the method may resist those attacks long enough to protect the information carried for a useful length of time.

Attacks are depends upon the many factors such as speed of the algorithm, key management .In some algorithms, if the size of the key is so huge it is impossible for an attacker to search through the key space with the resources they usually

International Journal of Computing Algorithm
Volume: 05 Issue: 01 June 2016, Page No. 29- 33
ISSN: 2278-2397

have . In some other algorithms, if the time (encryption/decryption) taken for the algorithm is less, the cipher cannot be broken. Depends on the study of possible attacks and performance of algorithm with respect to the key size, packet size, data type, data density and web browser , one can choose the algorithm which is Cryptographically strong .

## V. CONCLUSION

This paper provides the small description of various cryptographic algorithms and the methodology for analyzing performance of algorithms. In future, performance evaluation of selected cryptographic symmetric and asymmetric algorithms can be done by using these techniques to strengthen the security procedure and improve the speed.

**References**

[1] Cryptography in Practical UNIX and Internet Security http://www.diablotin.com/librairie / networking /puis/ch06_04.htm.
[2] Pratap Chandra Mandal "Evaluation of performance of the Symmetric Key Algorithms: DES, 3DES ,AES and Blowfish" in Journal of Global Research in Computer Science(Volume 3, No. 8, August 2012) ISSN-2229-371X.
[3] Ranjeet Masram, Vivek Shahare, Jibi Abraham, Rajni Moona "Analysis And Comparison Of Symmetric Keycryptographic Algorithms Based On Various File Features" in International Journal of Network Security & Its Applications in (IJNSA), Vol.6, No.4, July 2014.
[4] Rishabh Arora, Sandeep Sharma "Performance Analysis of Cryptography Algorithms" in International Journal of Computer Applications (0975 – 8887) Volume 48– No.21, June 2012.
[5] Sombir Singh1, Sunil K Maakar 2 and Dr. Sudesh Kumar "A Performance Analysis of DES and RSA Cryptography" International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)Web Site: www.ijettcs.org Email: editor@ijettcs.org, ditorijettcs@gmail.com Volume 2, Issue 3, May – June 2013 ( ISSN 2278-6856).
[6] G. Ramesh, R. Umarani Performance Analysis of Most Common Encryption Algorithms on Different Web Browsers in I.J. Information Technology and Computer Science, 2012, 12, 60-66 Published Online November 2012 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijitcs.2012.12.06.
[7] Lalit Singh, Dr. R.K. Bharti " Comparative Performance Analysis of Cryptographic Algorithms" International Journal of Advanced Research in Computer Science and Software Engineering in Volume 3, Issue 11, November 2013 (ISSN: 2277 128X).
[8] M.Anand kumar, S. Umadevi "Comparative analysis of symmetric encryption algorithms for data communication" in Karpagam JCS in Volume 7, Issue 5,July-Aug 2013(ISSN 0973-2926).
[9] G. Ramesh1 Dr. R. Umarani "Performance Analysis of Most Common Symmetrical Encryption Algorithms" International Journal of Power Control Signal and Computation(IJPCSC), Vol3. No1. Jan-Mar 2012 ISSN: 0976-268X www.ijcns.com.
[10] M.Anand kumar and K.Appathurai "Performance analysis of Blow Fish, IDEA and AES Encryption algorithms in Karpagam JCS in Volume 9, Issue 1,Nov-Dec 2014(ISSN 0973-2926).