International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 1, Issue 3, November 2014

A Comparative Study and Performance Evaluation of Cryptographic Algorithms: AES and Blowfish

K.Lakshmi Narayanan[1],P.Kannan[2], S.Esakki Rajavel[3]
Research Scholar, Department of Electronics and Communication Engineering, St.Peter's University, Chennai[1]
Assistant Professor, Department of ECE, Francis Xavier Engineering College, Tirunelveli[2,3]

Abstract: There are many aspects of security ranging from secure commerce and payments to private communications and protecting passwords. One essential aspect for secure communications is that of secret key Cryptography. It is the automated method in which security goals are accomplished. It includes the process of encryption that converts plain-text into cipher-text. The process of decryption reconverts the cipher-text into plain-text. Secure communication is the prime requirement of every organization. To achieve this, one can use many techniques or algorithms available for Cryptography. Various models were developed for the encryption in which keys were generated from the available data. ADVANCED ENCRYPTION STANDARD (AES) and BLOWFISH are commonly used for network data encryption. The main objective of this dissertation is to analyze encryption security, evaluated encryption speed and power consumption for both the algorithms. It is proved that the Blowfish encryption algorithm may be more suitable for wireless network application security.

Keywords: AES, Blowfish, cryptography, Information security.

I. INTRODUCTION

Security attacks against network are increasing significantly with time. Our communication media should also be secure and confidential. For this purpose, these three suggestions arrive in every one's mind: (i) one can transmit the message secretly, so that it can be saved from hackers, (ii) the sender ensures that the message arrives to the desired destination, and (iii) the receiver ensures that the received message is in its original form and coming from the right sender. For this, one can use two techniques, (i) one can use invisible ink for writing the message or can send the message through the confidential person, and (ii) one can use a scientific approach called "Cryptography".

Cryptography is the technique used to avoid unauthorized access of data. For example, data can be encrypted using a cryptographic algorithm in conjunction with the key management. It will be transmitted in an encrypted state, and later decrypted by the intended party. If a third party intercepts the encrypted data, it will be difficult to decipher. The security of modern cryptosystems is not based on the secrecy of the algorithm, but on the secrecy of a relatively small amount of information, called a secret key. The fundamental and classical task of cryptography is to provide confidentiality by encryption methods.

The encryption algorithms are usually summarized into two popular types: Symmetric key encryption and Asymmetric key encryption. Symmetric key algorithms are also called as secret key encryption, this symmetric key algorithm are a class of algorithms for cryptography that use trivially related, often identical, cryptographic keys for both encryption of plaintext and decryption of cipher text. The encryption key is trivially related to the decryption key, in that they may be identical or there is a simple transformation to go between the two keys. The keys, in practice, represent a secret between two or more parties that can be used to maintain a private information link. Other terms for symmetric key encryption are secret key, single key, shared key, one key, and private-key encryption. The representative Symmetric key cryptography algorithms include RC2, DES, 3DES, RC5, Blowfish, and AES, which use certain- or variable-length key.

Public-key cryptography, also known as asymmetric cryptography, is a form of cryptography in which the key used to encrypt a message differs from the key used to decrypt it. In public key cryptography, each user has a pair of cryptographic keys a public key and a private key. The private key is kept secret, while the public key may be widely distributed and used by other users. Incoming messages would have been encrypted with the recipient's public key and can only be decrypted with his corresponding private key. The keys are related mathematically, but the user private key cannot be derived from the widely used public key (E.g. RSA and Digital Signatures).

On the one hand, high security is the basic requirement of data encryption algorithm. On the other hand, encryption algorithms are known to be computationally intensive. They consume a significant amount of computing resources such as CPU time, memory, and battery power. Especially for a wireless device, usually with very limited resources (e.g. battery) is subject to the problem of energy consumption due to encryption algorithms. Therefore, it is essential to evaluate the performance of encryption algorithms so as to ensure various applications.

Both AES and blowfish algorithms are the most common cryptographic algorithm used for information security through wireless network, portable terminal, and so on. It is essential to evaluate their performance to ensure their domain application. It is also a significant work to facilitate the process of the encryption algorithm optimization.

In this paper, we firstly study the two most common encryption algorithm i.e. The AES and Blowfish encryption algorithm. Basically these algorithms are symmetric key encryption algorithms using block cipher. Referencing the encryption process methods, we analyze their security. We give a comprehensive performance evaluation which includes three aspects: security analysis, encryption speed, and power consumption. We design adequate experiment method for the evaluation. Based on the experimental results, we show the advantages and disadvantages for both encryption algorithms

II. Encryption Algorithm

In this section, we have an overview for the two encryption technique i.e. AES and Blowfish algorithms.

A. AES Encryption

Advanced Encryption Standard is the new encryption standard recommended by NIST to replace DES. It was originally called Rijndael (pronounced Rain Doll). It was selected in 1997 after a competition to select the best encryption standard. It has variable key length of 128, 192, or 256 bits; default 256. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices 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.

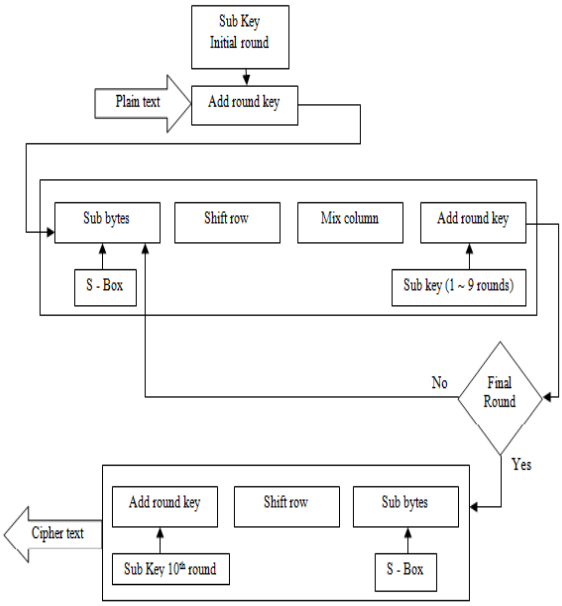

Fig 1: AES Encryption process

AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.

The Encryption and decryption process consists of a number of different transformations applied consecutively over the data block bits, in a fixed number of iterations, called rounds. The number of rounds depends on the length of the key used for the encryption process. For key length of 128 bits, the number of iteration required are10. (Nr = 10). As shown in fig 1 each of the first Nr-1 rounds consist of 4 operations: SubBytes (), ShiftRows (), MixColumns () & AddRoundKey ().

The four different transformations are described in detail below:

1) SubBytes (): It is a non-linear substitution of bytes that operates independently on each byte of the State using a substitution table (S box). This S-box which is invertible is constructed by first taking the multiplicative inverse in the finite field GF (2^8) with irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. The element {00} is mapped to itself. Then affine transformation is applied (over GF (2)).

2) ShiftRows (): Cyclically shifts the rows of the State over different offsets. The operation is almost the same in

the decryption process except for the fact that the shifting offsets have different values.

3) MixColumns (): This transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial a(x) = {03} x^3+ {01} x^2+ {02} x.

4) AddRoundKey (): In this transformation, a Round Key is added to the State by a simple bitwise XOR operation. Each Round Key consists of Nb words from the key expansion. Those Nb words are each added into the columns of the State. Key Addition is the same for the decryption process.

5) Key Expansion: Each round key is a 4-word (128-bit) array generated as a product of the previous round key, a constant that changes each round, and a series of S-Box lookups for each 32-bit word of the key. The Key schedule Expansion generates a total of Nb (Nr + 1) words.

The decryption process is direct inverse of the encryption process. All the transformations applied in encryption process are inversely applied to this process. Hence the last round values of both the data and key are first round inputs for the decryption process and follows in decreasing order.

B. Blowfish Encryption

In the early 1990's it was clear that Data Encryption Standard (DES) and even 3DES had begun to show its age. With the advancement of computer hardware it was possible to see the day in which DES would no longer be safe for sensitive data. In 1993 Bruce Schneider released his design of a successor to DES called Blowfish.

Characteristics of blowfish are as follows:

- It has block cipher of 64-bit block.
- The key length is variable and can be as long as 448 bits.
- It encrypts data on 32 bit microprocessors at a rate of 18 clock cycles per byte so much faster than AES, DES, and IDEA.
- Unpatented and royalty-free.
- It can run in less than 5K of memory.
- It has a simple structure and its implementation is easy.

The main claim to fame of Blowfish however is in its method of key scheduling. The round keys and the entire contents of all the S-boxes are created by multiple iteration of the block cipher. This enhances the security of the block cipher, since it makes exhaustive search of the key space very difficult, even for short keys.

C. Data Encryption

Encryption begins with a 64 bit block element of plain text that will be morphed into a 64 bit cipher text. The 64 bit segment is immediately split into two equally sized segments that will be used as the base of the Blowfish algorithm. The exclusive-or-operation (XOR) is performed between the first 32 bit block segment (L) and the first P array fig 2. The resulting 32 bit data is passed to the F function which permutes the data and yields a 32 bit block segment. This permuted block segment is XORed with the second 32 bit segment (R) created by the 64 bit plain text split. After the XOR operation is complete the 32 bit segments L and R are swapped for future iterations of the Blowfish algorithm.

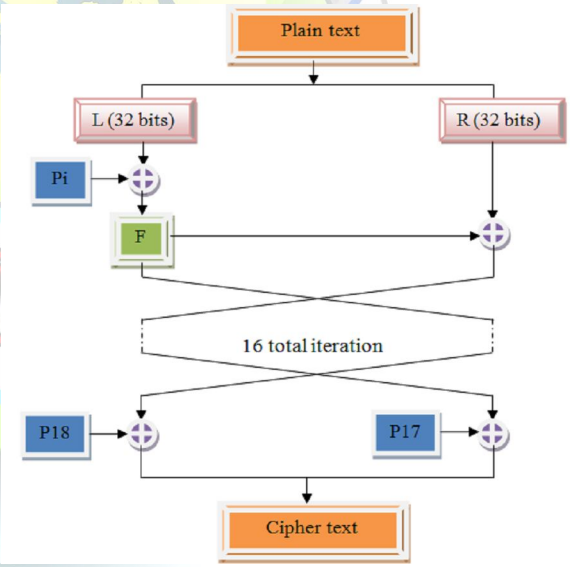

Fig 2: Blowfish encryption algorithm

D. Key Scheduling

Before traversal of the algorithm can begin, the P array and S-boxes must be defined. The P array is a reference to 18 independent sub arrays each of 32 bit length. Each P array and S-Box is initially defined using digits of π, which currently is assumed to not have a detectable pattern. The first P array (P1) is defined as the first 32 bits of π, the second P array (P2) is defined as the second 32 bits of π until all P arrays have been defined.

Blowfish allows an encryption key that ranges from 32 bits up to 448 bits. This algorithm is designed to accurately accept a variable key size by XOR the first 32 bits of the key with the first P array, the second 32 bits of the key, if present, with the second P array and continues until the end of the key schedule. If the end of the key is reached and P arrays are still waiting to be created the key rolls back to the first 32 bits and the execution continues. The resulting sub keys are still not considered to be truly cryptic although they come from random numbers.

*E. Feistel Structure of Blowfis*h

The F function is arguably the most complex section of the algorithm and the only section that uses the S-Boxes. The F function accepts a 32 bit stream of data and divides the input into four equal sections. Each 8 bit subdivision is transformed into a 32 bit data stream by means of their corresponding S-Box. The resulting 32 bit data is XORed or added together to provide a final 32 bit value for further permutations of the Blowfish algorithm fig 3.

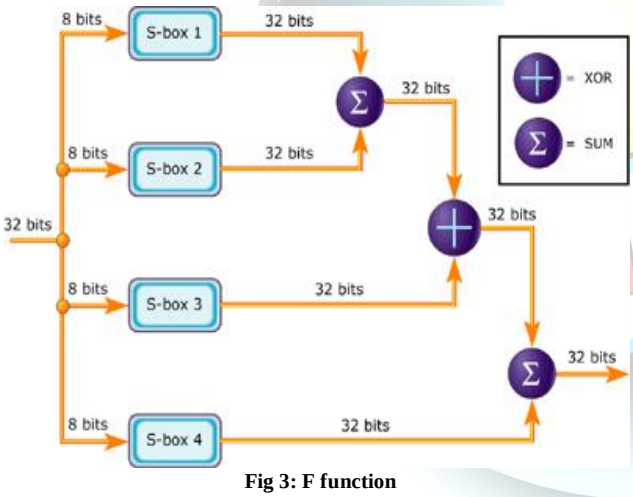

Fig 3: F function

III. Simulation Procedure

The main aim of this research is to evaluate the performance of this algorithm, so these algorithms were implemented in a uniform programming language. They were programmed in C and tested under windows XP operating system test platform is a Dell 14R laptop.

Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated by dividing the total plaintext in Megabytes encrypted on the total encryption time for each algorithm in. As the throughput value is increased, the power consumption of this encryption technique is decreased

By considering different sizes of data blocks (0.5MB to 20MB) the algorithms were evaluated in terms of the time required to encrypt and decrypt the data block. All the implementations were exact to make sure that the results will be relatively fair and accurate.

A. Simulation Result

Simulation results for this compassion point are shown in Fig. 4&5 and Table 1 at encryption decryption stages. The results show the superiority of Blowfish algorithm over other algorithms in terms of the processing time.

TABLE I
COMPARATIVE EXECUTION TIMES (IN MILLISECONDS) OF ENCRYPTION AND DECRYPTION ALGORITHMS WITH DIFFERENT PACKET SIZE

	Encryption		Decryption	
Input size in (Kbytes)	**Blowfish**	**AES**	**Blowfish**	**AES**
49	36	56	38	63
59	36	38	26	58
100	37	90	52	60
321	45	164	92	149
899	64	258	102	171
5345.28	122	1237	149	655
7310.33	107	1366	140	882
22335	155	1370	142	885
42000	165	1530	190	998
99000	190	1720	210	1208
Average Time (sec)	91.90	542.30	98.15	360.38
Throughput (Mb/sec)	152.25	25.90	142.58	38.83

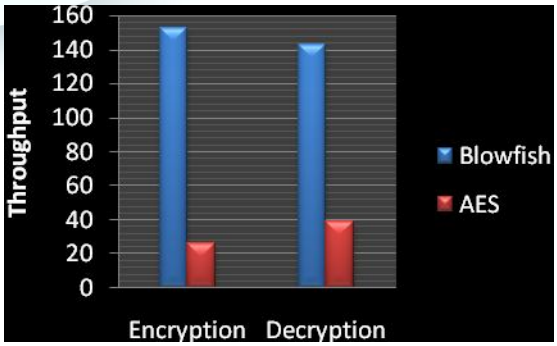

Fig 4: Throughput of AES and Blowfish algorithm (Mb/sec)

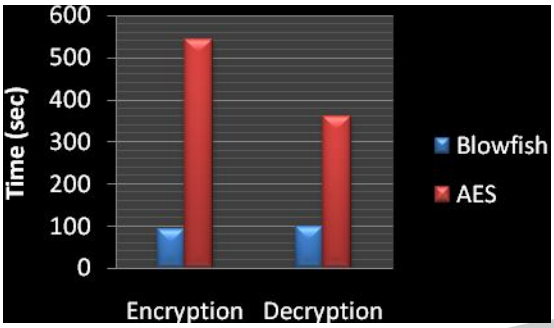

Fig 5: Time consumption comparison

IV. Modes Of Implementation

A. Electronic Code Book (Ecb) Mode

The simplest block cipher mode of operation is the electronic codebook mode (ECB) which is a block cipher mode of operation. One block of plain always produces the same block of cipher analysts learn that the block ``8d226acd" encrypts to the cipher-text block ``1c7ed351", they can immediately decrypt that cipher-text block whenever it appears in a message. This vulnerability is greatest at the beginning and end of messages, where well-defined headers and footers contain information about the sender, receiver, date, etc

B. Performance Result With Ecb Mode

The first set of experiments was conducted using (Electronic Code Book) ECB mode.

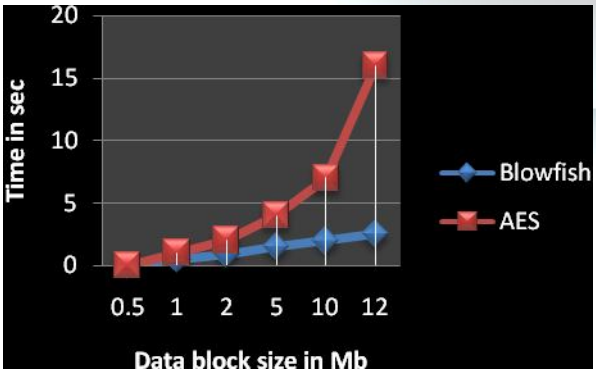

Fig 6: Performance result with ECB mode

The results show the superiority of Blowfish algorithm over AES algorithms in terms of processing time. It shows also that AES consumes more resources when data block size is relatively big.

C. Cipher Block Chaining (Cbc) Mode

Cipher Block Chaining (CBC) uses feedback to feed the result of encryption back into the encryption of the next block. The plain-text is XORed with the previous cipher-text block before it is encrypted. The encryption of each block depends on all the previous blocks. This requires that the decryption side processes all encrypted blocks sequentially. This mode requires a random initialization vector which is XORed with the first data block before it is encrypted. The initialization vector does not have to be kept secret. The initialization vector should0 be a random number (or a serial number), to ensure that each message is encrypted uniquely. An error in an encrypted block (caused by e.g. a transmission failure) causes the block with the error to be completely garbled. The subsequent block will have bit errors at the same positions as the original erroneous block. The blocks following the second block will not be affected by the error. Hence, CBC is self-recovering. While CBC recovers quickly from bit errors, it does not recover at all from synchronization errors. If a bit is added or lost from the cipher-text stream, then all subsequent blocks are garbled. A system that uses CBC must therefore ensure that the block structure remains intact. Like the ECB mode, also requires a complete block on its input before encryption can take place.

D. Perfomance Result With Cbc Mode

As expected, CBC requires more processing time than ECB because of its key-chaining nature. The results show in Fig.6 indicates also that the extra time added is not significant for many applications, knowing that CBC is much better than ECB in terms of protection.

The difference between two modes is hard to see by the naked eye, the results showed that the average difference between ECB and CBC is 0.059896 second, which is relatively small.

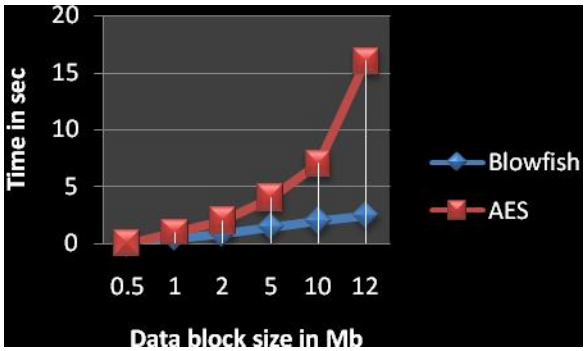

Fig 7. Performance result with CBC mode

V. CONCLUSION

Encryption algorithm plays an imperative task for information security guarantee in recent mounting internet and network application. In this paper, we studied two symmetric key encryption algorithms: AES and BLOWFISH. We assessed encryption speed, throughput and power burning up for their performance. The simulation results showed that Blowfish has superior performance than AES since Blowfish has not any known security weak points so far, it can be considered as an excellent standard encryption algorithm. BLOWFISH algorithm sprints faster than AES and showed poor performance results compared to BLOWFISH algorithms since it requires more processing power. Thus Blowfish algorithm may be more appropriate for wireless set-up which swaps small size packets.

REFERENCES

[1]. [FIPS 197, "Advanced Encryption Standard (AES)",November 26, 2011 http://csrc.nist.gov/publications /fips/fips197/fips-197.pdf

[2]. Bruce Schneier. "The Blowfish Encryption Algorithm Retrieved", October 25, 2008.

[3]. Simar Preet Singh, and Raman Maini "COMPARISON OF DATA ENCRYPTION ALGORITHMS" proceedings of International Journal of Computer Science and Communication Vol. 2, No. 1, January-June 2011, pp.125-127.

[4]. Aamer Nadeem, Dr M. Younus Javed "Aamer Nadeem, Dr M. Younus Javed" in the proceedings of IEEE pp 84-90,2010.

[5]. Daemen J., and Rijmen V., "AES proposal: Rijndael- The Rijndael Block Cipher", A Technical Report (Version 2) Presented to the National Institute of Standards and Technology (NIST), 2009.

[6]. W.S.Elkilani, "H.m.Abdul-Kader, "Performance of Encryption Techniques for Real Time Video Streaming, IBIMA Conference, Jan 2009, PP 1846-1850.

[7]. M.A. Viredaz and D.A. Wallach, "Power Evaluation of a Handheld Computer: A Case Study," WRL Research Report, 2011..

[8]. Wander, N. Gura, H. Eberle, V. Gupta, and S.Chang, Energy analysis for public-key cryptography for wireless sensor networks,. In IEEE PerCom'05, Pisa, Italy, Mar. 2010.

[9]. D. Salama, A. Elminaam and etal, "Evaluating the Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vo1.10, No.3,PP.216-222, May 2011.

[10]. P Verma, Ritu Agarwal, Dhiraj Dafouti, Shobha Tyagi "Peformance Analysis Of Data Encryption Algorithms" in the proceedings of IEEE, Pp399– 403, 2011.

[11]. S. Vaudenay, "On the Weak Keys in Blowfish," Fast Software Encryption, Third International Workshop Proceedings, Springer- Verlag, 2006, pp. 27-32.

[12]. P. Ruangchaijatupon, P. Krishnamurthy, "Encryption and Power Consumption in Wireless LANs-N,'' The ThirdIEEE Workshop on Wireless LANs - September 27-28, 2001-Newton, Massachusetts.

[13]. Karri, R. and Mishra, "Minimizing the secure wireless session energy," Journal of Mobile Network and Applications (MONET) 8, 2 (April), pp. 177-185.

[14]. J. Nechvatal et. al., Report on the development of Advanced Encryption Standard, NIST publication, Oct 2, 2000.

[15]. Guy-Armand Yandji, Lui Lian Hao, Amir-Eddine Youssouf, Jules Ehoussou, "RESEARCH ON A NORMAL FILE ENCRYPTION AND DECRYPTION" in proceedings of IEEE 2011.