Performance of Symmetric Encryption Algorithms

T. Lalitha[1] and Dr.R.Uma Rani[2]

[1]*Research Scholar, Centre for Research and Development, Bharathiyar University, Coimbator.*
lalithasrilekha@rediffmail.com
[2]*Asso. Professor, Computer Science, Department of Computer Science, Sri Saradha College for Women, Salem*
umainweb@gmail.com

Abstract- Encryption algorithms play a main role in information security systems. On the other side, those algorithms consume a significant amount of computing resources such as CPU time, memory, and battery power. This paper provides evaluation of six of the most common encryption algorithms namely: AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6. A comparison has been conducted for those encryption algorithms at different settings for each algorithm such as different sizes of data blocks, different data types ,battery power consumption, different key size and finally encryption/decryption speed.

Keywords – Symmetric encryption, Algorithms, Cryptography.

I. INTRODUCTION

Encryption is increasingly used to protect digital information, from personal details held on a computer to financial details transmitted over the Internet. Symmetric cryptography [2] uses the same key for both encryption and decryption. Using symmetric cryptography, it is safe to send encrypted messages without fear of interception (because an interceptor is unlikely to be able to decipher the message); however, there always remains the difficult problem of how to securely transfer the key to the recipients of a message so that they can decrypt the message.A major advance in cryptography occurred with the invention of public-key cryptography. The primary feature of public-key cryptography is that it removes the need to use the same key for encryption and decryption. With public-key cryptography, keys come in pairs of matched “public” and “private” keys. The public portion of the key pair can be distributed in a public manner without compromising the private portion, which must be kept secret by its owner. An operation done with the public key can only be undone with the corresponding private key.

Prior to the invention of public-key cryptography, it was essentially impossible to provide key management for large-scale networks. With symmetric cryptography, as the number of user’s increases on a network, the number of keys required to provide secure communications among those users increases rapidly. For example, a network of 100 users would require almost 5000 keys if it used only symmetric cryptography. Doubling such a network to 200 users increases the number of keys to almost 20,000. Thus, when only using symmetric cryptography, key management quickly becomes unwieldy even for relatively small-scale networks.

A. Encryption and digital signature

1) *Securing the electronic version* The simplest electronic version of the check can be a text file, created with a word processor, asking your bank to pay someone a specific sum. However, sending this check over an electronic network poses several security problems:

□ since anyone could intercept and read the file, you need confidentiality.

- since someone else could create a similar counterfeit file, the Bank needs to authenticate that it was actually you who created the file.
- since you could deny creating the file, the bank needs non-repudiation.
- since someone could alter the file, both you and the bank need data integrity.

2) *Digital signature*: The process of digitally signing starts by taking a mathematical summary (called a hash code) of the check. This hash code is a uniquely-identifying digital fingerprint of the check. If even a single bit of the check changes, the hash code will dramatically change. The next step in creating a digital signature is to sign the hash code with your private key. This signed hash code is then appended to the check.How is this a signature? Well, the recipient of your check can verify the hash code sent by you, using your public key. At the same time, a new hash code can be created from the received check and compared with the original signed hash code. If the hash codes match, then the recipient has verified that the check has not been altered. The recipient also knows that only you could have sent the check because only you have the private key that signed the original hash code.

3) *Confidentiality and encryption:*Once the electronic check is digitally signed, it can be encrypted using a high-speed mathematical transformation with a key that will be used later to decrypt the document. This is often referred to as a symmetric key system because the same key is used at both ends of the process. As the check is sent over the network, it is unreadable without the key. The next challenge is to securely deliver the symmetric key to the bank.

4) *Public-key cryptography for delivering symmetric keys:* Public-key encryption [3] is used to solve the problem of delivering the symmetric encryption key to the bank in a secure manner. To do so, you would encrypt the symmetric key using the bank's public key. Since only the bank has the corresponding private key, only the bank will be able to recover the symmetric key and decrypt the check Why use this combination of public-key and symmetric cryptography? The reason is simple. Public-key cryptography is relatively slow and is only suitable for encrypting small amounts of information – such as symmetric keys. Symmetric cryptography is much faster and is suitable for encrypting large amounts of information.

II. ENCRYPTION ALGORITHMS

Many encryption algorithms are widely available and used in information security. They can be categorized into Symmetric (private) and Asymmetric (public) keys encryption. In Symmetric keys encryption or secret key encryption, only one key is used to encrypt and decrypt data. The key should be distributed before transmission between entities. Keys play an important role. If weak key is used in algorithm then every one may decrypt the data. Strength of Symmetric key encryption depends on the size of key used. For the same algorithm, encryption using longer key is harder to break than the one done using smaller key. There are many examples of strong and weak keys of cryptography algorithms like RC2, DES, 3DES, RC6, Blowfish, and AES. RC2 uses one 64-bit key, DES uses one 64-bits key. Triple DES (3DES) uses three 64- bits keys while AES uses various (128,192,256) bits keys.Blowfish uses various (32-448); default 128bits while RC6 is used various (128,192,256) bits keys [1-5].

Asymmetric key encryption or public key encryption is used to solve the problem of key distribution. In Asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption (E.g. RSA and Digital Signatures). Because users tend to use two keys: public key, which is known to the public and private key which is known only to the user. There is no need for distributing them prior to transmission. However, public key encryption is based on mathematical functions, computationally intensive and is not very efficient for small mobile devices [1]. Asymmetric encryption techniques are almost 1000 times slower than Symmetric techniques, because they require more computational processing power.

B. *Brief definitions of the most common encryption techniques are given as follows:*

1) *DES* : (Data Encryption Standard), was the first Cryptography encryption standard [4] to be recommended by NIST (National Institute of Standards and Technology).DES is (64 bits key size with 64 bits block size) . Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher [3][4].

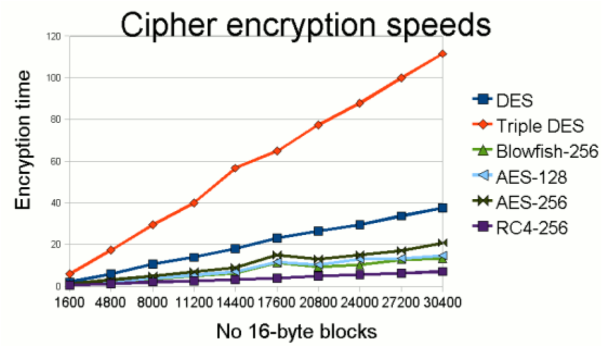

Figure 1: The time taken to encrypt various numbers of 16-byte blocks of data using the algorithms mentioned.

2) *3DES* is an enhancement of DES; it is 64 bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. It is a known fact that 3DES is slower than other block cipher methods [3].

3) *RC2: RC2* is a block cipher with a 64-bits block cipher with a variable key size that range from 8 to128 bits. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts [3].

4) *Blowfish[5]* is block cipher 64-bit block - can be used as a replacement for the DES algorithm. It takes a variable length key, ranging from 32 bits to 448 bits; default 128 bits. Blowfish is unpatented, license-free, and is available free for all uses. Blowfish has variants of 14 rounds or less. Blowfish is successor to Twofish [5].

5) *AES[7]* is a block cipher .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 encryption is fast and flexible; it can be implemented on various platforms especially in small devices [6]. Also, AES has been carefully tested for many security applications [3], [7].

6) *RC6* is block cipher derived from RC5. It was designed to meet the requirements of the Advanced Encryption Standard competition. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits. Some references consider RC6 as Advanced Encryption Standard [7].

Table 1: Characteristics of commonly used encryption algorithms

Algorithm	Key size(s)	Speed	Speed depends on key size?	Security / comments
RC4	40-1024	Very fast	No	Of questionable security; may be secure for moderate numbers of encrypted sessions of moderate length.
Blowfish	128-448	Fast	No	Believed secure, but with less attempted cryptanalysis than other algorithms.
AES	128, 192, 256	Fast	Yes	It has the advantage of allowing a 256-bit key size, which should protect against certain future attacks
DES	56	Slow	–	Insecure
Triple DES	112/168, but equivalent security of 80/112	Very slow	No	Triple DES performs three DES operations (encrypt-decrypt-encrypt), using either two or three different keys.

III PROPERTIES OF SYMMETRIC KEY ENCRYPTION

A. *Correctness Property:*It ensures that if message is encrypted using a key, then the same message is returned as the output when the cipher text is decrypted i.e.

If c ←Enc(m,K) then

m ← Dec(c,K) ∀ m € M, K €2 KeySpace

B. *Security Property*: We study the properties an SE possesses in order to be securely used for achieving confidentiality. In other words, we study various notions of security of an SE.The first property that we study is called perfect or unconditional or information-theoretic secrecy. In achieving perfect secrecy, we consider an adversary who has an unbounded computational power (i.e it can take infinite amount of computing power to break an SE).

Definition 1 (Perfectly Secure Encryption.) A symmetric encryption is said to be perfectly secure if,

Pr(m = m’) = Pr(m = m’|c = c’) ∀ m € M; c € C; c’ = Enc(m’,K)

Intuitively, this means that in a perfectly secure encryption, the knowledge of cipher text is as good as no knowledge of it.

Definition 2 (Perfectly Secure Encryption.) A symmetric encryption is said to be perfectly secure if,

Pr(Enc(m1,k) = c) = Pr(Enc(m2,k) = c) m1,m2 € M, c € C

Intuitively, this means that all the messages are equally likely to be the plaintext corresponding to a given cipher text.The two definitions can be shown to be equivalent. In other words,

Definition1 ↔Definition2

One Time Pad (OTP): A Perfectly Secure Encryption Scheme. This encryption scheme provides perfectly secrecy. The message space and key space are of same size. It consists of:

c) Key Generation Algorithm: A randomized algorithm that outputs a random key.

- Input : Null

- $K \xleftarrow{\$} \text{KeyGen}()$

IV. ALGORITHM TYPES AND STRENGTHS

Let’s take a closer look at both symmetric and public key cryptography. As a subset of cryptography, cryptographic algorithms can be divided into two categories:

1) *Stream algorithms* :Operate on plaintext one byte at a time, where a byte is a character, number, or special character. The process is inefficient and slow.

2) *Block algorithms* **–** Operate on plaintext in groups of bytes, called blocks (hence the name block algorithms or block ciphers). Typical block sizes for modern algorithms is 64 bytes, small enough to work with but large enough to deter code breakers. Unfortunately, with the current speed of microprocessors, breaking a 64-byte algorithm using brute force is proving to be to relatively easy task [4]. Encrypted information is only as good as the key required to decrypt it. In theory, any key can eventually be obtained and the encrypted information decrypted successfully. It’s really a question of the time required to break the key. In the early years of cryptography,before the computer, even small keys were nearly impossible to break. The advantage gained with computers, however, is pure speed.

Public key algorithms [6] can use much larger keys to achieve secrecy. Where breaking symmetric keys is usually done through brute force, public key algorithms involve deriving the matching secret key from the public key. The process depends on the kind of encryption adopted but generally involves the use of prime factors. 768-byte keys are safe for the near future and 1028-byte keys are safe for the foreseeable future, whereas 256-byte keys can be easily cracked by personal computers [5:1].

V. CONCLUSION

Encryption is one of a number of tools that can be used to safeguard electronic information and privacy.Encryption tools are widely available and are becoming more sophisticated; the government is encouraging their uptake both for private users and for businesses. Availability of encryption tools means that the government faces a challenge in encouraging its legal use whilst ensuring that it is not misused by criminals. There is a need for users to monitor advances in encryption technology continually to ensure electronic data is adequately protected.

VI. REFERENCES

[1] Ruangchaijatupon, P. Krishnamurthy, "Encryption and Power Consumption in Wireless LANs-N,’’ The Third IEEE Workshop on Wireless LANs - September 27-28, 2001- Newton, Massachusetts.

[2] Hardjono, "Security In Wireless LANS And MANS," Artech House Publishers 2005.

[3] W.Stallings, "Cryptography and Network Security 4th Ed," Prentice Hall , 2005,PP. 58-309 .

[4] Coppersmith, D. "The Data Encryption Standard (DES) and Its Strength Against Attacks."I BM Journal of Research and Development, May 1994,pp. 243 -250.

[5] Bruce Schneier. The Blowfish Encryption Algorithm Retrieved October 25, 2008, http://www.schneier.com/blowfish.html

[6] K. Naik, D. S.L. Wei, Software Implementation Strategies for Power-Conscious Systems,” Mobile Networks and Applications - 6, 291-305, 2001.

[7] Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard."D r. Dobb's Journal, March 2001,PP. 137-139.