DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis

Бате Жожи

International Journal of Emerging Technology and …

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 1, Issue 2, December 2011)

DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis

Jawahar Thakur[1], Nagesh Kumar[2]
[1] Department of Computer Science, Himachal Pradesh University, Shimla, INDIA
[2]Department of Computer Science, Himachal Pradesh University, Shimla, INDIA

[1]jawahar.hpu@gmail.com
[2]engg.nagesh2@gmail.com

Abstract- Security is the most challenging aspects in the internet and network applications. Internet and networks applications are growing very fast, so the importance and the value of the exchanged data over the internet or other media types are increasing. Hence the search for the best solution to offer the necessary protection against the data intruders' attacks along with providing these services in time is one of the most interesting subjects in the security related communities. Cryptography is the one of the main categories of computer security that converts information from its normal form into an unreadable form. The two main characteristics that identify and differentiate one encryption algorithm from another are its ability to secure the protected data against attacks and its speed and efficiency in doing so.

This paper provides a fair comparison between three most common symmetric key cryptography algorithms: DES, AES, and Blowfish. Since main concern here is the performance of algorithms under different settings, the presented comparison takes into consideration the behavior and the performance of the algorithm when different data loads are used. The comparison is made on the basis of these parameters: speed, block size, and key size. Simulation program is implemented using Java programming.

Keywords: Cryptography, DES, AES, Blowfish, Encryption, Decryption.

I. INTRODUCTION

Cryptography is usually referred to as "the study of secret". Encryption is the process of converting normal text to unreadable form. Decryption is the process of converting encrypted text to normal text in the readable form.

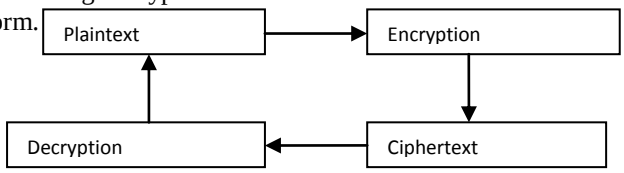

Figure 1: Conventional Encryption Model

Steps involved in the conventional encryption model:

- A sender wants to send a Hello message to a recipient.
- The original message, also called plaintext, is converted to random bits known as ciphertext by using a key and an algorithm. The algorithm being used can produce a different output each time it is used, based on the value of the key.
- The ciphertext is transmitted over the transmission medium.
- At the recipient end, the ciphertext is converted back to the original text using the same algorithm and key that was used to encrypt the message. Figure 1 below shows the conventional cryptographic process.

As defined in RFC 2828 [11], cryptographic system is "a set of cryptographic algorithms together with the key management processes that support use of the algorithms in some application context." The definition gives the whole mechanism that provides the necessary level of security comprised of network protocols and data encryption algorithms.

A. Cryptography Goals:

There are five main goals of cryptography.Every security system must provide a bundle of security functions that can assure the secrecy of the system. These functions are usually referred to as the goals of the security system. These goals can be listed under the following five main categories (Earle, 2005):

- Authentication: The process of proving one's identity. This means that before sending and receiving data using the system, the receiver and sender identity should be verified.
- Privacy/confidentiality: Ensuring that no one can read the message except the intended receiver. Usually this function is how most people identify a secure system.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 1, Issue 1, November 2011)

- It means that only the authenticated people are able to interpret the message content and no one else.
- Integrity: Assuring the receiver that the received message has not been altered in any way from the original. The basic form of integrity is packet check sum in IPv4 packets.
- Non-repudiation: A mechanism to prove that the sender really sent this message. Means that neither the sender nor the receiver can falsely deny that they have sent a certain message.
- Service Reliability and Availability: Since secure systems usually get attacked by intruders, which may affect their availability and type of service to their users. Such systems provide a way to grant their users the quality of service they expect.

B. Symmetric and Asymmetric Encryptions:

There are two main categories of cryptography depending on the type of security keys used to encrypt/decrypt the data. These two categories are: Asymmetric and Symmetric encryption techniques.

1) Symmetric Encryption

It is also called as single key cryptography. It uses a single key. In this encryption process the receiver and the sender has to agree upon a single secret (shared) key. Given a message (called plaintext) and the key, encryption produces unintelligible data, which is about the same length as the plaintext was. Decryption is the reverse of encryption, and uses the same key as encryption.

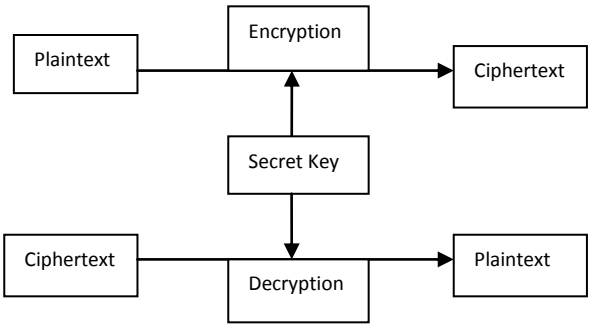

Figure 2: Symmetric Key Cryptography Process

2) Asymmetric Encryption

It is also called as public key cryptography. It uses two keys: public key,which is known to the public, used for encryption and private key, which is known only to the user of that key, used for decryption. The public and the private keys are related to each other by any mathematical means. In other words, data encrypted by one public key can be encrypted only by its corresponding private key. Encryption and decryption procedure as shown below in figure 3:

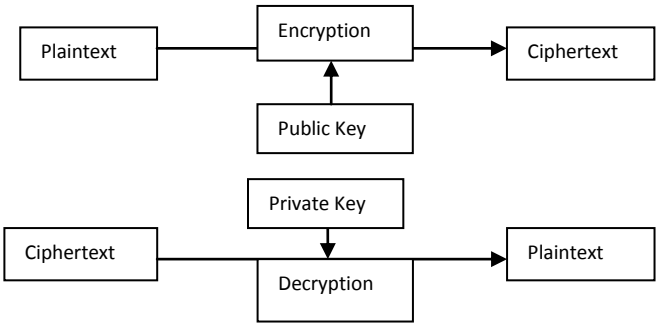

Figure 3: Public Key Cryptography Process

C. Modes of Encryption/Decryption

1) ECB (Electronic Code Book)

In this mode data is divided into 64-bit blocks and each block is encrypted one at a time. Separate encryptions with different blocks are totally independent of each other. This means that if data is transmitted over a network or phone line, transmission errors will only affect the block containing the error. It also means, however, that the blocks can be rearranged, thus scrambling a file beyond recognition, and this action would go undetected. ECB is the weakest of the various modes because no additional security measures are implemented besides the basic DES algorithm. However, ECB is the fastest and easiest to implement, making it the most common mode of DES seen in commercial applications. This is the mode of operation used by Private Encryptor.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 1, Issue 1, November 2011)

2) CBC (Cipher Block Chaining)

In this mode of operation, each block of ECB encrypted ciphertext is XORed with the next plaintext block to be encrypted, thus making all the blocks dependent on all the previous blocks. This means that in order to find the plaintext of a particular block, you need to know the ciphertext, the key, and the ciphertext for the previous block. The first block to be encrypted has no previous ciphertext, so the plaintext is XORed with a 64-bit number called the Initialization Vector, or IV for short. So if data is transmitted over a network or phone line and there is a transmission error (adding or deleting bits), the error will be carried forward to all subsequent blocks since each block is dependent upon the last. If the bits are just modified in transit (as is the more common case) the error will only affect all of the bits in the changed block, and the corresponding bits in the following block. The error doesn't propagate any further. This mode of operation is more secure than ECB because the extra XOR step adds one more layer to the encryption process.

3) CFB (Cipher Feedback)

In this mode, blocks of plaintext those are less than 64 bits long can be encrypted. Normally, special processing has to be used to handle files whose size is not a perfect multiple of 8 bytes, but this mode removes that necessity (Private Encryptor handles this case by adding several dummy bytes to the end of a file before encrypting it). The plaintext itself is not actually passed through the DES algorithm, but merely XORed with an output block from it, in the following manner: A 64-bit block called the Shift Register is used as the input plaintext to DES. This is initially set to some arbitrary value, and encrypted with the DES algorithm. The ciphertext is then passed through an extra component called the M-box, which simply selects the left-most M bits of the ciphertext, where M is the number of bits in the block we wish to encrypt. This value is XORed with the real plaintext, and the output of that is the final ciphertext. Finally, the ciphertext is fed back into the Shift Register, and used as the plaintext seed for the next block to be encrypted. As with CBC mode, an error in one block affects all subsequent blocks during data transmission. This mode of operation is similar to CBC and is very secure, but it is slower than ECB due to the added complexity.

4) OFB (Output Feedback)

This is similar to CFB mode, except that the ciphertext output of DES is fed back into the Shift Register, rather than the actual final ciphertext. The Shift Register is set to an arbitrary initial value, and passed through the DES algorithm. The output from DES is passed through the M-box and then fed back into the Shift Register to prepare for the next block. This value is then XORed with the real plaintext (which may be less than 64 bits in length, like CFB mode), and the result is the final ciphertext. Note that unlike CFB and CBC, a transmission error in one block will not affect subsequent blocks because once the recipient has the initial Shift Register value; it will continue to generate new Shift Register plaintext inputs without any further data input. However, this mode of operation is less secure than CFB mode because only the real ciphertext and DES ciphertext output is needed to find the plaintext of the most recent block. Knowledge of the key is not required.

Section 2 will give a brief review of all the concerned research papers. It will provide a brief discussion of the other contributors and their conclusions. Section 3 will discuss the main objective of research. Section 4 will discuss the methodology used in the work with simulation settings. Section 5 will give the results of the research and provide discussion about the same. Finally, section 6 concludes this paper by summarizing the key points and other related considerations.

II. Background Study

A. Compared Algorithms:

DES: (Data Encryption Standard), was the first encryption standard to be published by NIST (National Institute of Standards and Technology). It was designed by IBM based on their Lucifer cipher. DES became a standard in 1974 (www.tropsoft.com). DES uses a 56 bit key, and maps 64 bit input block into a 64 bit output block. The key actually looks like a 64 bit quantity, but one bit in each of the 8 octets is used for odd parity on each octet. There are many attacks and methods recorded till now those exploit the weaknesses of DES, which made it an insecure block cipher.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 1, Issue 1, November 2011)

AES: (Advanced Encryption Standard), also known as the Rijndael(pronounced as Rain Doll) algorithm, is a symmetric block cipher that can encrypt data blocks of 128 bits using symmetric keys 128, 192, or 256. AES was introduced to replace the DES. Brute force attack is the only effective attack known against this algorithm.

Blowfish: Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Blowfish is unpatented and license-free, and is available free for all uses. Though it suffers from weak keys problem, no attack is known to be successful against it (Bruce, 1996) (Nadeem, 2005).

B. Other Contributions

(Tamimi, 2008) provided a performance comparison between four most common algorithms: DES, 3DES, AES, and Blowfish. The comparison had been conducted by running several different settings to process different sizes of data blocks to evaluate the algorithm's encryption/decryption speed. The simulation setup was in C# programming language. The results of this paper shows that blowfish has a better performance than other common encryption algorithms. AES showed poor performance results compared to other algorithms since it requires more processing power.

(Nadeem, 2005) In this paper, the popular secret key algorithms including DES, 3DES, AES (Rijndael), Blowfish, were implemented, and their performance was compared by encrypting input files of varying contents and sizes. The algorithms were implemented in Java programming, using their standard specifications, and were tested on two different hardware platforms, to present the comparison. The two different machines are: P-II 266 MHz and P-IV 2.4 GHz.

(Dhawan, 2002) has also done experiments for comparing the performance of the different encryption algorithms implemented inside .NET framework. Their results are close to the ones shown before. The comparison was performed on the following algorithms: DES, Triple DES (3DES), RC2 and AES (Rijndael).

The results shows that AES outperformed other algorithms in both the number of requests processes per second in different user loads, and in the response time in different user-load situations.

(N. Penchalaiahet.al., 2010) discussed the principal advantages of AES with respect to DES, as well as its limitations. They said that AES can be quite comfortably implemented in high level or low level languages.

(Elminaam et. al., 2010) presented a comparison of AES, DES, 3DES, RC2, Blowfish and RC6. They used 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. They concluded that in case of changing packet size Blowfish showed better performance than other algorithms followed by RC6.

AES had better performance than RC2, DES, and 3DES. In case of changing key size – it was concluded that higher key size leads to clear change in the battery and time consumption.

(Singhal and Raina, 2011) presented a comparative analysis between AES and RC4 for better utilization. In this paper authors tried to find out performance comparison between block ciphers (AES) and stream cipher (RC4) algorithm. Based on the analysis and result, this paper concluded that which algorithm is better to use based on different performance metrics. The various metrics were: Encryption time, Decryption time, Throughput, CPU process time, Memory Utilization.

III. Objective

The objective of the paper is to provide a performance analysis between symmetric key cryptography algorithms: DES, AES and Blowfish. The analysis has been conducted by running several encryption settings to process different sizes of data blocks to evaluate the algorithm's speed for encryption and decryption. The paper also shows the analysis on the basis of two block cipher modes: ECB, CBC, OFB, and CFB. Each algorithm is designed and executed in these modes.The variation is provided in data size given by the user. The data is retrieved from various text files to calculate the time consumed by each algorithm to process the retrieved data.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 1, Issue 1, November 2011)

IV. Methodology

A. Simulation and Settings:

The simulation uses the provided classes in java environment to simulate the performance of DES, AES and Blowfish. The implementation uses managed wrappers for DES, AES and Blowfish available in java.cypto and java.security[CryptoSpec] that wraps unmanaged implementations available in JCE (Java Cryptography Extension) & JCA (Java Cryptography Architecture).The Cipher class provides the functionality of a cryptographic cipher used for encryption and decryption. It forms the core of the JCE framework.

Table 1: Algorithms' Settings

Algorithm	Key Size(Bits)	Block Size(Bits)
DES	64	64
AES	128	128
Blowfish	128	64

The evaluation is meant to evaluate the results by using block ciphers. Hence, the load data (plaintext) is divided into smaller block size as per algorithm settings given in Table 1 above.

B. System Parameters:

The experiments are conducted using AMD Sempron processor with 2GB of RAM. The simulation program is compiled using the default settings in jdk 1.7development kit for JAVA. The experiments will be performed couple of times to assure that the results are consistent and are valid to compare the different algorithms.

C. Experiment Factors:

Since the security features of each algorithm as their strength against cryptographic attacks is already known and discussed. The chosen factor here to determine the performance is the algorithm's speed to encrypt/decrypt data blocks of various sizes.

V. Results and Analysis

This section will show the results which are obtained by running the simulation program using different data loads. The results show the impact of changing data load on each algorithm and the impact of Cipher Mode used.

A. Performance Results with ECB:

The first set of experiments were conducted using ECB mode, the results are shown in figure 4 below. The results show the superiority of Blowfish algorithm over other algorithms in terms of the processing time. It shows also that AES consumes more resources when the data block size is relatively big.

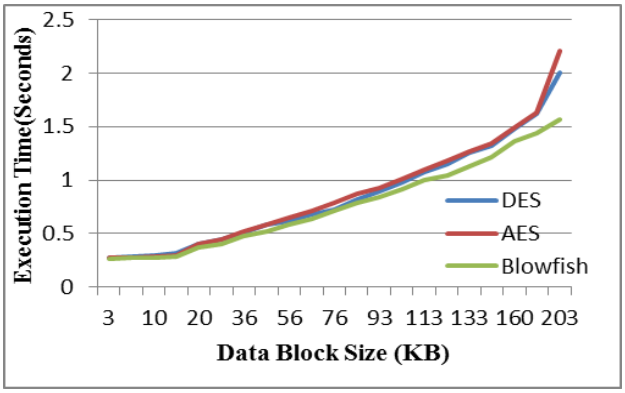

Figure 4: Performance Results with ECB mode

B. Performance Results with CBC:

The second set of experiments were conducted using CBC mode, the results are shown in figure 5 below.

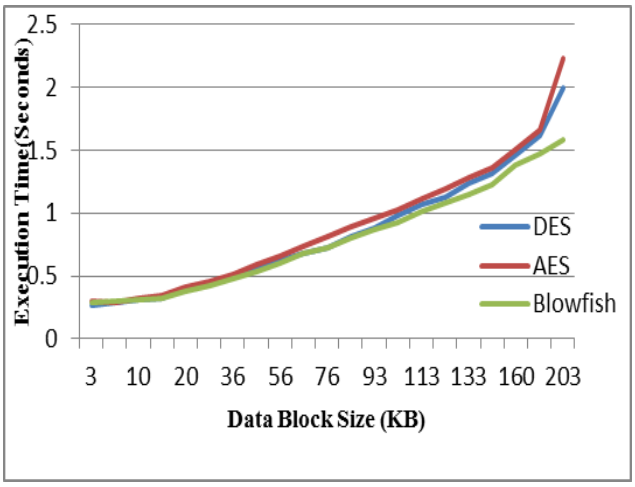

Figure 5: Performance Results with CBC mode

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 1, Issue 1, November 2011)

As expected CBC require more processing time than ECB because of its key-chaining nature. The results indicate 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 the two modes is hard to see by the naked eye because it is relatively small. Again the results show the superiority of Blowfish algorithm over other algorithms in terms of the processing time.

C. Performance with OFB Mode:

The third set of experiments were conducted using OFB mode, the results are shown in figure 6 below. As expected OFB require less processing time than ECB & CBC. The results indicate that the OFB is better for applications requiring output feedback. The difference between the three modes is hard to see by the naked eye because it is relatively small. Again the results show the superiority of Blowfish algorithm over other algorithms in terms of the processing time.

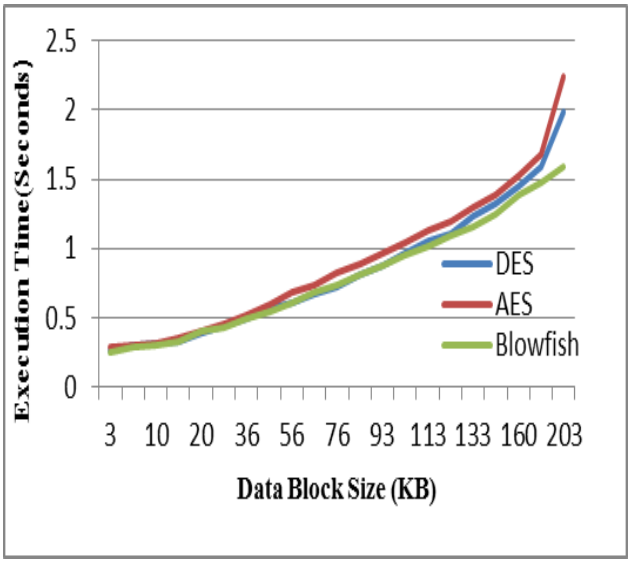

Figure 6: Performance Results with OFB mode

D. Performance with CFB:

The fourth set of experiments were conducted using CFB mode, the results are shown in figure 7 below. As expected CFB require less processing time than ECB & CBC. The results indicate that the OFB is better than CFB in terms of processing time. The difference between the four modes is hard to see by the naked eye because it is relatively small. The results show the superiority of Blowfish algorithm over other algorithms in terms of the processing time.

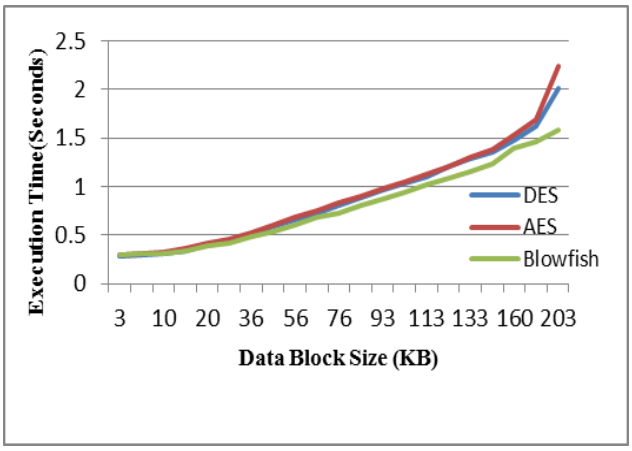

Figure 7: Performance Results with CFB mode

VI. Conclusion and Future Scope

The presented simulation results showed that Blowfish has a better performance than other common encryption algorithms used. Since Blowfish has not any known security weak points so far, this makes it an excellent candidate to be considered as a standard encryption algorithm. AES showed poor performance results compared to other algorithms since it requires more processing power. Using CBC mode has added extra processing time, but overall it was relatively negligible especially for certain application that requires more secure encryption to a relatively large data blocks. OFB shows better performance than ECB and CBC but require more processing time than CFB. Overall time differences between all modes are negligible.

In future this analysis can be implemented in better simulators to get better results. This analysis can be done in another simulator by taking networking into consideration to show which algorithm performs better in network. The simulators which can be used are: MATLAB, ns2, ns3, OPNET, NetSim etc. These simulators will give better results for cryptographic applications in network.

References:

[1] Singhal, Nidhi and Raina, J P S. "Comparative Analysis of AES and RC4 Algorithms for Better Utilization", International Journal of Computer Trends and Technology, ISSN: 2231-280, July to Aug Issue 2011, pp. 177-181.

[2] Singh, S Preet and Maini, Raman. "Comparison of Data Encryption Algorithms", International Journal of Computer Science and Communication, vol. 2, No. 1, January-June 2011, pp. 125-127.

[3] Elminaam, D S Abd; Kader H M Abdual and Hadhoud, M Mohamed. "Evaluating the Performance of Sysmmetric Encryption Algorithms", International Journal of Network Security, Vol. 10, No. 3, pp. 216-222, May 2010.

[4] Penchalaiah, N. and Seshadri, R. "Effective Comparison and Evaluation of DES and Rijndael Algorithm (AES)", International Journal of Computer Science and Engineering, Vol. 02, No. 05, 2010, 1641-1645.

[5] Stallings, W; "Cryptography and Network Security: Principals and Practices", Prentice Hall, 8th Edition, 2009.

[6] Tamimi, A Al; "Performance Analysis of Data Encryption Algorithms", Oct 2008.

[7] Results of Comparing Tens of Encryption Algorithms Using Different Settings – Crypto++ Benchmarks, Retrieved Oct. 1, 2008. (http://www.eskimo.com/weidai/benchmarks.html).

[8] Nadeem, Aamer; "A Performance Comparison of Data Encryption Algorithms", IEEE 2005.

[9] [Hardjono2005]," Security In Wireless LANS And MANS ,". Artech House Publishers 2005.

[10] Stallings, W;"Cryptography and Network Security", Prentice Hall, 4th Edition, 2005.

[11] [Edney2003]," Real 802.11 Security: Wi-Fi Protected Access and 802.11i ,". Addison Wesley 2003.

[12] [RFC2828],"Internet Security Glossary", http://www.faqs.org/rfcs/rfc2828.html.

[13] [TropSoft] "DES Overview", http://www.tropsoft.com/strongenc/des.htm.

[14] Dhawan, Priya; "Performance Comparison: Security Design Choices," Microsoft Developer Network October 2002. http://msdn2.microsoft.com/en-us/library/ms978415.aspx

[15] [Bruce1996] Schneier, Bruce; "Applied Cryptography", John Wiley & Sons, Inc 1996.

[16] WEBSITE http://download.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec

[17] National Institute of Standards and Technology, Data Encryption Standard, FIPS 46-2, 1993.

[18] Kaufman, Charlie; Perlman, Radia and Speciner, Mike. "Network Security Private Communication in a Public World"; Second Edition; Pearson Education; Prentice Hall.

[19] Stallings, William; "Cryptography and Network Security Principles and Practices"; Fourth Edition; Pearson Education; Prentice Hall; 2009.

[20] Stinson, D.; "Cryptography, Theory and Practice"; CRC Press; Second edition; 2000.

[21] Menezes, A., Oorschot, P. and Vanstone, S. (1996). "Handbook of Applied Cryptography".CRC Press.

[22] Moshopoulos, Nikosand and Chaniotakis, Eleftherios; "A Survey of Cryptography Algorithms – Trends and Products"; National Technical University of Athens, Electrical & Computer Engineering Department, HeroonPolytehneiou 9, 15773 Zographou, Athens, GREECE.

[23] Schneier, B.; Kelsey, J.; Whiting,D.; Wagner, D.; Hall C. and Ferguson N.; "Performance Comparison of the AES Submissions"; version 2.0; 1999.

[24] Jorstad, Norman D.;"Cryptographic Algorithm Metrics "; Institute for Defense Analyses Science and Technology Division; 1997.

[25] Verisign: http://www.verisign.com.

[26] Rivest, R. L., Shamir, A., Adelmann, L.: "A method for obtaining digital signature and public –key cryptosystems", Commun. ACM, 1978, VOL. 21, pp. 120-126

[27] Jorstad, Norman D.; "Cryptographic Algorithm Metrics "; Institute for Defense Analyses Science and Technology Division; 1997.

[28] Gustafson, H.; et al.; "A Computer Package for Measuring the Strength of Encryption Algorithms,"; Computers & Security, Vol 13, No. 8, 1994, Elsiver Science, Ltd., pp. 687-697.

[29] Towbridge, Dave; "Public-key Crypto Gives Privacy Power To The People,"; Computer Technology Review, Vol XV, No. 4, April 1995, p 10.

[30] Lenstra, A. K. and Verheul, E. R.; "Selecting cryptographic key sizes,"; Journal of Cryptography 14(4), 2001, 255-293.

[31] "A Brief History of Cryptography"; 2001. University of Dayton School of Law. 19 Oct. 2004 <http://cybercrimes.net/Cryptography/Articles/Hebert.html >.

[32] "History of Cryptography" Wikipedia. <http://en.wikipedia.org/wiki/History_of_cryptography>.

[33] www.cc.gatec