Comparison of DES, AES, Blowfish and Twofish Symmetric Key Cryptography Algorithms

[1]R.venkateshwarlu
[1]Associate Professor-Department of Computer Science &Engineering,
[1]Jyothishmathi Institute of Technology& Science, Karimnagar, Telangana State, India 505481.

Abstract-Currently, security is the most effective hand of Internet and network applications. In the current generation, Internet applications and networks are growing very fast, so the importance and value of data exchanged through the Internet or other types of media are increasing. Therefore, the search for the best solution to offer the necessary protection against illegal data attacks along with the provision of these services on time is one of the most interesting topics in communities related to security. Cryptography is one of the main categories of computer security that converts information from its readable form into an illegible form. The two main characteristics that identify and differentiate one encryption algorithm from another are its ability to protect data protected against attacks and its speed and efficiency to make it so important.
This document provides a fair comparison between the four most common symmetric key cryptography algorithms: DES, AES, Blowfish and Twofish. Since the main concern here is the performance of the algorithms in different configurations, the presented comparison takes into account the behavior and performance of the algorithm when different data loads are used. The comparison is made based on these parameters: speed, block size and key size. A simulation program is implemented using C programming and Java programming.

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

1. 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.
Cryptography is generally known as "the study of secrecy". Encryption is the process of converting normal text to an illegible form. Decryption is the process of converting encrypted text to normal text in readable form.

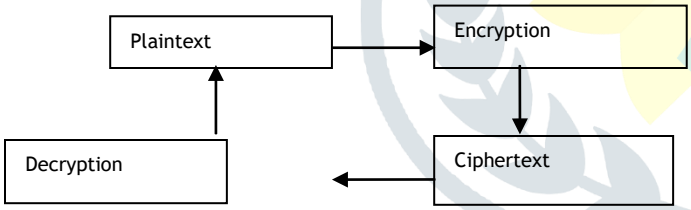

Figure 1: Conventional Encryption and Decryption Model

Steps involved in the conventional encryption model.

• A sender wishes to send a global greeting message to a recipient.
• The original message, also called plain text, is converted to random bits known as encrypted text by using a key and an algorithm. The algorithm that is used can produce a different output each time it is used, depending on the value of the key.
• Encrypted text is transmitted through the transmission medium.
• At the end of the recipient, the encrypted text is converted back to the original text using the same algorithm and key used to encrypt the message. Figure 1. Next, the conventional cryptographic process is shown.
As defined in RFC 2828 [11], the cryptographic system is "a set of cryptographic algorithms together with key management processes that support the use of algorithms in some application context". The definition provides the entire mechanism that provides the necessary level of security comprising network protocols and data encryption algorithms.

I. Objectives of cryptography.

There are five main goals of cryptography. Each security system must provide a set of security functions that can guarantee the confidentiality of the system. These functions are often referred to as security system objectives. These objectives can be listed in 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 the gender identity must be verified.
• Privacy / confidentiality: ensure that nobody can read the message, except the recipient. Generally, this function is the way in which most people identify a safe system.

• It means that only authenticated people are able to interpret the content of the message and no one else.
• **Integrity:** assuring the recipient that the received message has not been altered in any way with respect to the original. The basic form of integrity is a checksum of packets in IPv4 packets.
• **Non-repudiation:** a mechanism to prove that the sender actually sent this message. It means that neither the sender nor the recipient can falsely deny that they have sent a certain message.
• **Reliability and availability of the service:** since secure systems are usually attacked by intruders, which can affect their availability and the type of service for their users. These systems provide a way to give their users the quality of service they expect.
II.Symmetric and Asymmetric encryption.

There are two main categories of cryptography according to the type of security keys used to encrypt / decrypt the data. These two categories are asymmetric and symmetric encryption techniques.

a) Symmetric encryption

It is also called a single-key cryptography. Use only one key In this encryption process, the receiver and the sender must agree on a secret (shared) key. Given a message (called plain text) and the key, the encryption produces unintelligible data, which is approximately the same length as plain text. The decryption is the back of the encryption and uses the same key as the encryption.

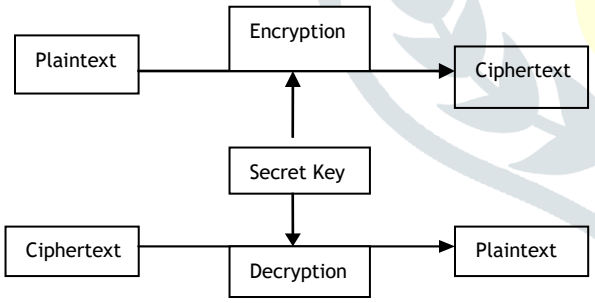

Figure 2: Symmetric Key Cryptography Process

b).AsymmetricEncryption

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

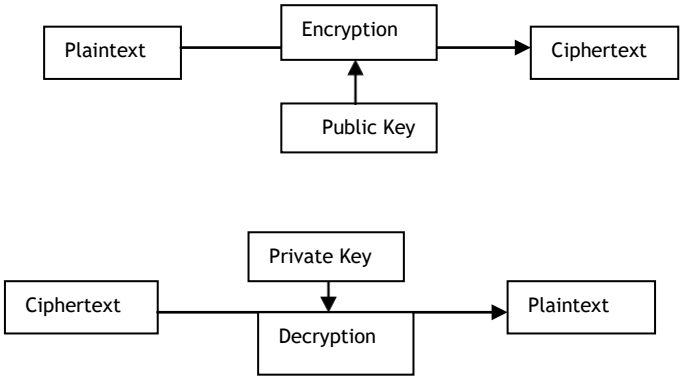

Figure 3: Public Key CryptographyProcess

III. Encryption / Decryption modes

1.ECB (Electronic Code Book)

In this mode, the data is divided into blocks of 64 bits and each block is encrypted one by one. Separate encryptions with different blocks are completely independent of each other. This means that if the data is transmitted over a network or telephone line, transmission errors will only affect the block containing the error. However, it also means that blocks can be rearranged, therefore, by mixing a file beyond recognition, and this action will not be detected. The ECB is the weakest of the various modes because no additional security measures are implemented in addition to the basic DES algorithm. However, the ECB is the fastest and easiest to implement, making it the most common DES mode seen in commercial applications. This is the operating mode used by Private Encryptor.

2.CBC (Cipher Block Chaining).

In this mode of operation, each block of text encrypted with ECB is XORed with the next block of plain text to be encrypted, so that all blocks depend on all the previous blocks. This means that to find the simple text of a particular block, you need to know the encrypted text, the key and the cipher text of the previous block. The first block to be encrypted does not have a previous encryption text, so the plain text is in XOR with a 64-bit number called Initialization Vector, or IV for short. Therefore, if the data is transmitted through a network or telephone line and there is a transmission error (add or delete bits), the error will be transferred to all subsequent blocks, since each block depends on the last ones. If the bits are modified in transit (as is the most common case), the error will only affect all the bits in the modified block and the corresponding bits in the next block. The error does not spread anymore. This mode of operation is more secure than the ECB because the additional XOR step adds one more layer to the encryption process.

3.CFB (Cipher Feedback).

In this mode, the plain text blocks are less than64 bits long can be encrypted. Normally, special processing must be used to handle files whose size is not a perfect multiple of 8 bytes, but this mode eliminates that need (Private Encryptor handles this case by adding several fictitious bytes to the end of a file before encrypting it). The plain text itself is not actually passed through the DES algorithm, but is simply XORed with an output block, as follows: a 64-bit block called Shift Register is used as plain text input to DES. This is initially set to an arbitrary value and is encrypted with the DES algorithm. The encrypted text is then passed through an additional component called M-box, which simply selects the M bits to the left of the cipher text, where M is the number of bits in the block that we wish to encrypt. This value is in XOR with the real simple text, and the output of that is the final cipher text. Finally, the encrypted text is fed back into the shift register and used as the plain text seed for the next block to be encrypted. As with CBC mode, an error in a block affects all subsequent blocks during data transmission. This mode of operation is similar to the CBC and is very safe, but it is slower than the ECB due to the added complexity.

4.OFB (Output Feedback).

This is similar to CFB mode, except that the encrypted text output of DES is returned to the change log, instead of the actual final cipher text. The Shift Register is set to an arbitrary initial value and passed through the DES algorithm. The output of the DES is passed through the M box and then returned to the shift register to prepare the next block. This value is then in XOR with the actual simple text (which may be less than 64 bits in length, such as CFB mode), and the result is the final cipher text. Note that, unlike CFB and CBC, a transmission error in a block will not affect the subsequent blocks because once the recipient has the value of the initial change register; will continue to generate new text entries without registration of shift records without any other data entry. However, this mode of operation is less secure than the CFB mode because only the actual encrypted text and the output of the DES encrypted text are needed to find the plain text of the most recent block. No knowledge of the key is required.Section 2 will give a brief review of all the research work in question. It will provide a brief discussion of the other contributors and their conclusions. Section 3 will discuss the main objective of the investigation. Section 4 will discuss the methodology used in the work with simulation settings. Section 5 will give the results of the investigation and provide a discussion about it. Finally, section 6 concludes this document by summarizing the key points and other related considerations.

2. Background Study

A. Compared Algorithms.

DES: (Data Encryption Standard), was the first encryption standard published by the NIST (National Institute of Standards and Technology). It was designed by IBM based on its Lucifer cipher. DES became a standard in 1974. DES uses a 56-bit key and assigns a 64-bit input block to a 64-bit output block. The key really looks like a 64 bit amount, but a bit is used in each of the 8 octets for the odd parity in each octet. There are many attacks and methods registered so far that exploit the weaknesses of DES, which made it an insecure block cipher.

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

Blowfish: Blowfish is a symmetric block cipher that can be used effectively for encryption and data protection. It takes a key of variable length, of 32 bits to 448 bits, reason why it is ideal to assure the data. Blowfish was designed in 1993 by Bruce Schneier as a quick and free alternative to existing encryption algorithms. Blowfish is not patented and is not licensed, and is available free of charge to all users. Although he has a weak-keys problem, no attack is known to be successful against him (Bruce, 1996) (Nadeem, 2005).

Twofish: Twofish is a 128-bit symmetric key block cipher proposed by B.Schneier [19]. Twofish accepts a variable length key of up to 256 bits. The encryption is a Feistel network of 16 rounds that adapted components from the Khufu [20], Square [21] and SAFER [22] ciphers. Notable features of the Twofish design include S-dependent key boxes, separable maximum distance matrices (MDS), pseudo-Hadamard transform (PHT) and a highly complex key program. Basic terms used in Twofish.

B. Other contributions

(Tamimi, 2008) provided a performance comparison between the four most common algorithms: DES, 3DES, AES, Blowfish and Twofish. The comparison was made by executing several different configurations to process different sizes of data blocks to evaluate the encryption / decryption speed of the algorithm. The simulation configuration was in the C # programming language. The results of this document show that puffer fish has a better performance than other common encryption algorithms. AES showed poor performance results compared to other algorithms, since it requires more processing capacity.

(Nadeem, 2005) In this document, the popular secret key algorithms including DES, 3DES, AES (Rijndael), Blowfish, Twofish were implemented and their performance was compared by encrypting input files of different contents and sizes. The algorithms were implemented in Java and Java programming, using their standard specifications, and 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 conducted experiments to compare the performance of the different encryption algorithms implemented within the .NET framework. Their results are close to those shown above. The comparison was made in the following algorithms: DES, Triple DES (3DES), RC2, RC4, RC5, RC6 and AES (Rijndael).

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

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

(Elminaam et al., 2010) presented a comparison of AES, DES, 3DES, RC2, RC4, RC5, RC6, Blowfish and Twofish. They used different configurations for each algorithm, such as different sizes of data blocks, different types of data, battery power consumption, different key sizes and, finally, the encryption / decryption speed. They came to the conclusion that in the case of changing the size of the Blowfish package, Twofish showed a better performance than other algorithms followed by RC6.

AES had a better performance than RC2, RC4, RC5, RC6, DES and 3DES. In case of changing the size of the key, it was concluded that a larger size of the key leads to a clear change in the battery and the consumption of time.

(Singhal and Raina, 2011) presented a comparative analysis between AES and RC4 for better use. In this article, the authors attempted to find out the performance comparison between block ciphers (AES) and the flow cipher algorithm (RC6). Based on the analysis and the result, this document concluded that the algorithm is best used based on different performance metrics. The various metrics were: Encryption time, Decryption time, Performance, CPU processing time, Memory utilization.

3. OBJECTIVE

The aim of the article is to provide a performance analysis between symmetric key cryptography algorithms: DES, AES Blowfish and Twofish. The analysis was performed by executing various encryption configurations to process different sizes of data blocks to evaluate the speed of the algorithm for encryption and decryption. The document also shows the analysis based on two block encryption modes: BCE, CBC, OFB and CFB. Each algorithm is designed and executed in these modes. The variation is provided in the size of the data given by the user. The data is retrieved from several text files to calculate the time consumed by each algorithm to process the recovered data.

4. METHODOLOGY

A. Simulation and adjustments.

The simulation uses the classes provided in the Java environment to simulate the performance of DES, AES, Blowfish and Twofish. The implementation uses managed wrappers for DES, AES, Blowfish and Twofish available in java.cypto and java.security [CryptoSpec] that involves unmanaged implementations available in JCE (Java Cryptography Extension) and JCA

(Java Cryptography Architecture). The Cipher class provides the functionality of a cryptographic encryption used for encryption and decryption. It forms the core of the JCE framework.

A. Performance results with the ECB.

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

Table 1: Algorithms' Settings

Algorithm	Key Size(Bits)	Block Size(Bits)
DES	64	64
AES	128	128
Blowfish	128	64
Twofish	128	256

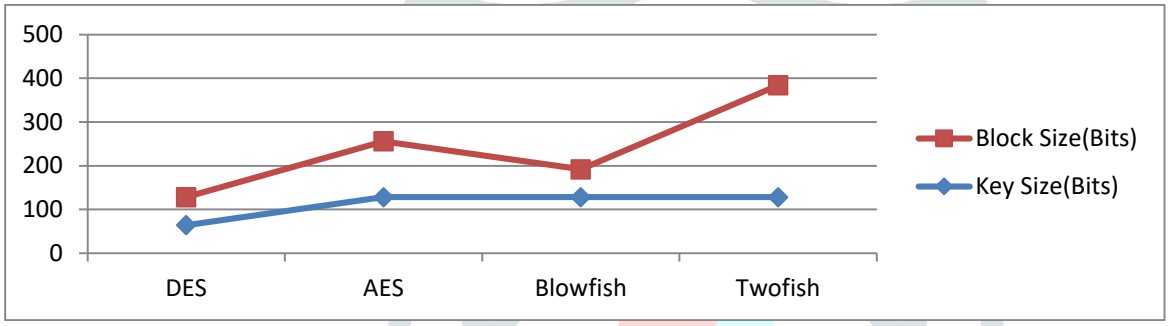

Figure 4: Performance Results with ECB mode

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 are given in Table 1 above.

The objective of the evaluation is to evaluate the results using block ciphers. Therefore, the load data (plain text) is divided into a smaller block size, according to the algorithm configurations given in Table 1 above.

B. System parameters.

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

C. Factors of the experiment:

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

5. RESULTS AND ANALYSIS.

This section will show the results obtained when executing the simulation program using different data loads. The results show the impact of changing the data load in each algorithm and the impact of the encryption mode used.

B. Performance results with CBC.
The second set of experiments was performed using the CBC mode, the results are shown in figure 5 below.

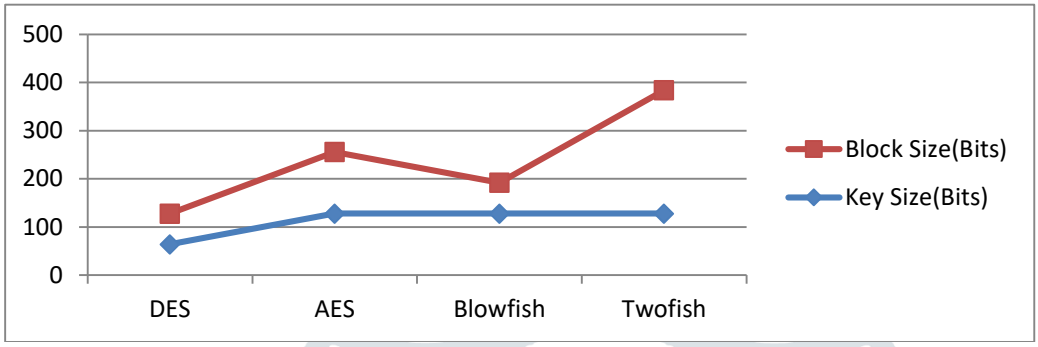

Figure 5: Performance Results with CBC mode

As expected, CBCs require more processing time than the ECB due to their nature of key chaining. The results indicate that the additional time added is not significant for many applications, knowing that CBC is much better than the ECB in terms of protection. The difference between the two modes is difficult to see with the naked eye because it is relatively small. Again, the results show the superiority of the Blowfish and Twofish algorithms over other algorithms in terms of processing time.
C. A performance with the OFB mode.
The third set of experiments was performed using the OFB mode, the results are shown in figure 6 below. As expected, OFB requires less processing time than ECB and CBC. The results indicate that the OFB is better for applications that require output feedback. The difference between the three modes is difficult to see with the naked eye because it is relatively small. Again, the results show the superiority of the Blowfish algorithm over other algorithms in terms of processing time.

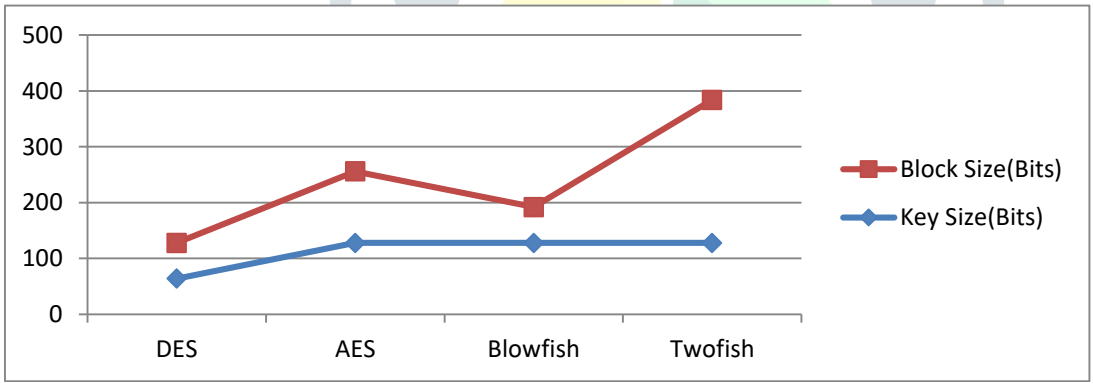

Figure 6: Performance Results with OFB mode

D. Performance with CFB.

The fourth set of experiments was performed using the CFB mode, the results are shown in figure 7 below. As expected, CFBs require less processing time than the ECB and the CBC. The results indicate that the OFB is better than the CFB in terms of processing time. The difference

between the four modes it is difficult to see with the naked eye because it is relatively small. The results show the superiority of Blowfish, Twofish algorithms over other algorithms in terms of processing time.

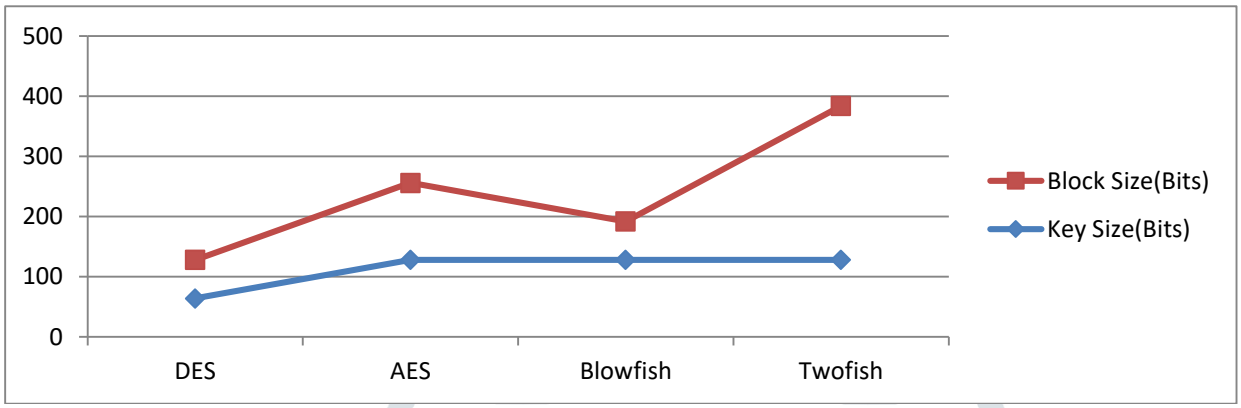

Figure 7: Performance Results with CFB mode

6. CONCLUSION AND FUTURE SCOPE

The results of the simulation presented showed that Twofish has a better performance than other common encryption algorithms used. As Twofish does not have any known security weaknesses 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 capacity. The use of CBC mode has added an additional processing time, but in general it was relatively insignificant, especially for certain applications that require a more secure encryption for relatively large data blocks. OFB shows better performance than ECB and CBC, but it requires more processing time than CFB. The global time differences between all modes are negligible.

In the future, this analysis can be implemented in better simulators to obtain better results. This analysis can be done in another simulator taking into account the networks to show which algorithm works best in the network. The simulators that can be used are MATLAB, ns2, ns3, OPNET, NetSim, etc. These simulators will give better results for cryptographic applications in the network.

References:

[1] Singhal, Nidhi and Raina, JP S. "Comparative analysis of the AES and RC4 algorithms for better use", International Journal of Trends and Computer Technology, ISSN: 2231-280, July to August, 2011 edition, pp . 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 Abdul and Hadhoud, M Mohamed. "Performance Evaluation of Symmetric 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 the Rijndael Algorithm (AES)", International Journal of Computer Science and Engineering, vol. 02, No. 05, 2010, 1641-1645.

[5] Jams, W; "Cryptography and network security: principles and practices", Prentice Hall, 8th edition, 2009.

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

[7] Results of the comparison of dozens of encryption algorithms that use different configurations: Crypto ++ reference points, obtained on October 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 networks LANS AND MANS
, "Artech House Publishers 2005.

[10] Jams, 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 Options," 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/s 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. "Private communication of security in networks in a public world" Second edition; Pearson Education; Prentice Hall.

[19] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson, "Twofish: A 128-Bit Block Cipher", 1998, [online] Available at http://www.certainkey.com/ resources /article/twofish.pdf

[20] R. Merkle. "Software fast encryption functions". In A.J. Menezes and S.A.Vanstone, editors, Advances in Cryptology - CRYPTO'90, LNCS 537, pp. 476 ~ 501. Springer Verlag, 1991.

[21] J. Daemen, L. Knudsen, and V. Rijmen. "The encrypted block of the square". In E. Bi-ham, editor, Fast Software Encryption, Fourth International Workshop, Haifa, Israel, January 1997, LNCS 1267, p. 149 ~ 165. Springer Verlag, 1997.

[22] J.L. Massey. "SAFER K-64: A byte-oriented block encryption algorithm". In R. Anderson, editor, Fast Software Encryption - Proc. Cambridge Security Workshop, Cambridge, U.K., LNCS 809, pgs. 1 ~ 17. Springer Verlag, 1994.

[23] Jams, William; "Principles and practices of cryptography and network security"; Fourth edition; Pearson Education; Prentice Hall; 2009

[24] Stinson, D .; "Cryptography, theory and practice"; CRC Press; Second edition; 2000

[25] Menezes, A., Oorschot, P. and Vanstone, S. (1996). "Manual of applied cryptography" .CRC Press.

[26] Moshopoulos, Nikos and Chaniotakis, Eleftherios; "A survey of cryptography algorithms - Trends and products"; National Technical University of Athens, Department of Electrical and Computer Engineering, HeroonPolytehneiou 9, 15773 Zographou, Athens, GREECE.

[27] Schneier, B .; Kelsey, J .; Whiting, D .; Wagner, D .; Hall C. and Ferguson N .; "Performance comparison of AES presentations"; version 2.0; 1999.

[28] Jorstad, Norman D .; "Cryptographic algorithm metrics"; Institute of Defense Analysis, Division of Science and Technology; 1997

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

[30] Rivest, R. L., Shamir, A., Adelmann, L.: "A method to obtain digital signatures and cryptosystems of public key", Common. ACM, 1978, VOL. 21, pp. 120-126

[31] Jorstad, Norman D .; "Metrics of the cryptographic algorithm"; Institute of Defense Analysis Division of Science and Technology; 1997

[32] Gustafson, H .; et al .; "A computer package to measure the strength of encryption algorithms"; Computers & Security, Vol.

13, No. 8, 1994, Elsevier Science, Ltd., pp. 687-697.

[33] Trowbridge, Dave; "The Crypto public key gives people privacy power"; Journal of Computer Technology, Vol. XV, No. 4, April 1995, p. 10
[34] Lenstra, A.K., and Verheul, E.R .; "Cryptographic key size selection"; Cryptography Journal 14 (4), 2001, 255-293.

[35] "A brief history of cryptography"; 2001. Law School of the University of Dayton. October 19, 2004
http://cybercrimes.net/Cryptography/Articles/Hebert.html

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

[37] www.cc.gatec