Performance comparison of 3DES, AES, Blowfish and RSA for Dataset Classification and Encryption in Cloud Data Storage

Daniel Commey
Department of Computer Engineering
Kwame Nkrumah University of Science and Technology (KNUST)

Selorm Griffith Klogo
Department of Computer Engineering
Kwame Nkrumah University of Science and Technology (KNUST)

James Dzisi Gadze
Department of Telecommunications Engineering
Kwame Nkrumah University of Science and Technology (KNUST)

ABSTRACT

Cloud data storage are logical pools containing physical storages on multiple servers in different locations. All over the world, organizations and individuals are leveraging the storage of their confidential data to cloud data storages to reduce cost and enjoy the numerous benefits the cloud has to offer. Nonetheless, there are several security risks associated with outsourcing data to cloud data storages and the most concerning among these risks is the breach of data confidentiality. Data confidentiality is crucial in cloud data storages as the impact risk of a data breach can be catastrophic. Encryption is the best way to ensure data confidentiality and that, data is protected from unauthorized disclosure. The two most important features of encryption algorithms are the guarantee of security without any vulnerabilities (strength of encryption) and the efficiency (performance) of the algorithm. This paper presents a survey of popular encryption algorithms (Triple DES, AES, Blowfish and RSA) and their performance comparison report by simulating the selected encryption algorithms with a classification framework.

General Terms

Security, Algorithms, Cloud Computing, Data Storage

Keywords

Encryption, Encryption Algorithms, Cloud Data Storage, Data Confidentiality

1. INTRODUCTION

Encryption is widely used to safeguard cloud data privacy. It includes data transformation (encryption) using mathematical techniques and a secret key, called the encryption key. Only with the use of a decryption key can encrypted data be disclosed to approved parties [1]. Encryption prevents an attacker from easily getting access to sensitive data when stored in a cloud data centre or transiting over the network. Two kinds of encryption techniques are available: symmetric and asymmetric [2]–[4].

Symmetric cryptography utilizes the same keys to encrypt and decrypt the information, making the algorithms less complicated than their asymmetric counterparts [3]. Their performance is, therefore, better than asymmetric cryptography algorithms, usually 100 to 1000 times faster. The safety level depends on the key's length. The National Standards and Technology Institute (NIST) of the United States recommends 160–512 bits [4]. Usually, symmetric cryptography is used for bulk data encryption and is used in protocols such as Internet Protocol Security (IPSec) and Transport Layer Security (TLS). Nonetheless, secure distribution of keys is a challenge in symmetric cryptography [4], [5].

Asymmetric cryptography utilizes two distinct keys: an encryption private key and a decryption public key [5]. For instance, assume Ama intends to use asymmetric cryptography to send a secret message to Kofi. Ama first uses the public key of Kofi to encrypt the plaintext message, which is public and available to anyone [6]. In the same way, Kofi can make his public key accessible in his organization's folder or on a public page. However, Kofi also has a private key that is known to him alone, and different from his public key [5]. These keys are complementary in function despite being distinct because Kofi will use his private key to decrypt the message from Ama [4]. In other words, only the corresponding private key can decrypt data in plaintext that is encrypted with a public key. Asymmetric cryptography is used as a system for implementing digital signatures to overcome the main allocation or key distribution challenge in symmetric cryptography [3], [4].

1.1 Rivest-Shamir-Adleman (RSA) Encryption Algorithm

RSA was designed in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman. For key exchange, digital signatures or information block encryption, it is one of the best-recognized public-key cryptosystems. RSA uses an encryption block of variable size and a key of variable size. It is a number-theory based asymmetric (public key) cryptosystem. It generates public and private keys using two prime numbers. For encryption and decryption purposes, these two distinct keys are used. The sender encrypts the message using the public key and the receiver can decrypt it using his own private key when the message is transmitted to the receiver. Figure 1 below depicts the RSA multiple blocks process.

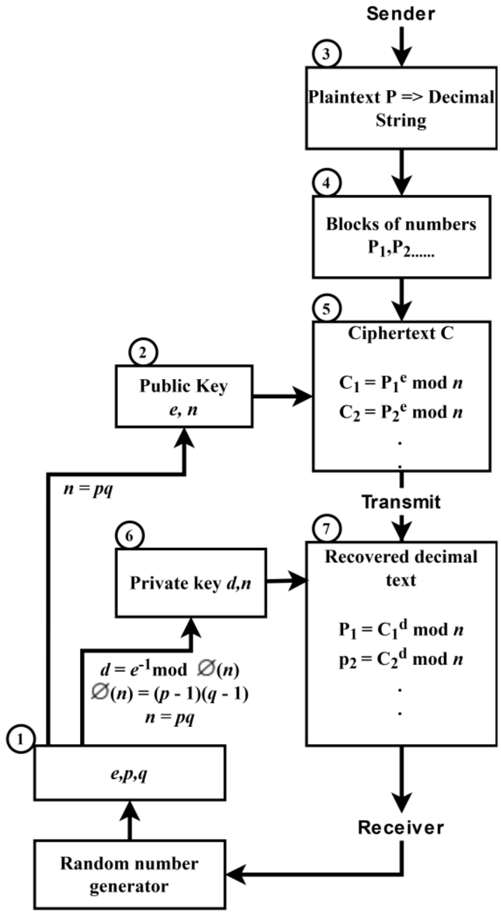


Figure 1: RSA Multiple blocks processing. Source: [7]

1.2 Triple Data Encryption Standard (3DES) Algorithm

As a result of developments in key searching, the triple-DES (3DES) algorithm was developed as a substitute for the Data Encryption Standard (DES). 3DES is a proposition based on the current DES and has been standardized for key management in ANSI X9.17 & ISO 8732 and in PEM [8]. It was also suggested by ANSI X9 for the overall EFT standard [9], [10]. It has backwards compatibility with current single DES (when K1=K2=K3). Either two or three *56-bit* keys are used by the 3DES algorithm. The efficient duration of the key is therefore up to *168 bits* [9].

3DES involves the use of three *64-bit* DES keys (***K1, K2, K3***) in Encrypt-Decrypt- Encrypt (EDE) mode, i.e. the plain text is encrypted with ***K1***, decrypted with ***K2***, then encrypted with ***K3*** again. Three key options are defined by the standard:

1. The preferred option uses three separate keys (***K1 ≠ K2 ≠ K3 ≠ K1***). It provides *3×56= 168 bits* keyspace.
2. The second option uses two separate keys and a third key that is the same as the first key (***K1 ≠ K2*** and ***K3 = K1***). This provides *2×56= 112 bits* keyspace.
3. The third option is a three-key main bundle (***K1 = K2 = K3***). This choice is equal to DES Algorithm.

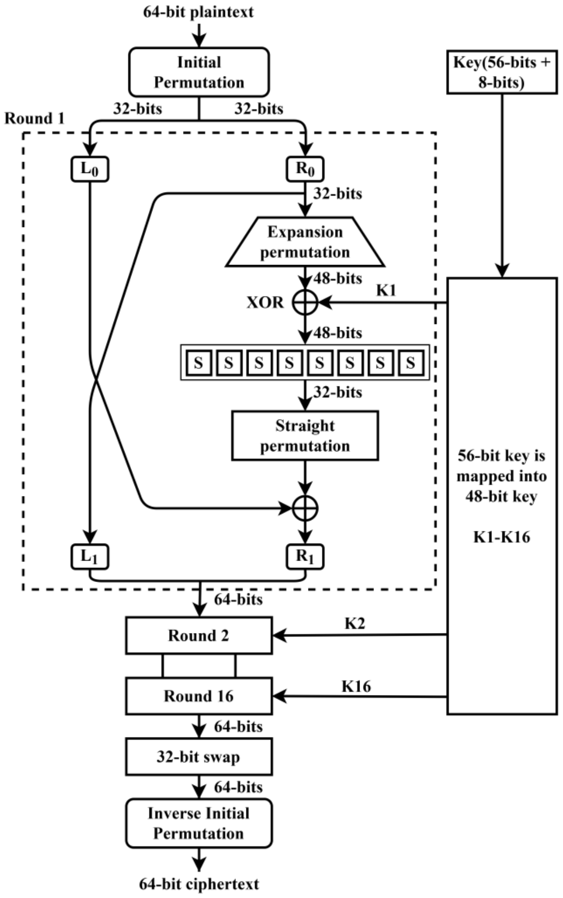


Figure 2: General Depiction of DES. Source: [8]

1.3 Advanced Encryption Standard (AES) Algorithm

NIST announced a call for applicant ciphers for its new Advanced Encryption Standard (AES) in September 1997, as it obviously needed a replacement for DES at the time [11]. The cipher candidates had to present their proposals by June 1998, and in October 2000 a finalist was selected. In June 1998, a total of 15 applicants were accepted (6 were rejected as unfinished), and in August 1999, 5 were shortlisted. Finally, in October 2000, Rijndael was selected as the finalist of the AES algorithm [12].

All submissions and unclassified analyses were published by NIST. AES algorithm was the latest generation of block ciphers and have a significant increase in block size-from the old *64-bit* to *128-bit* standard; and *128-bit* to *256-bit* keys. This was motivated in part by the public protests of DES and RC-5 exhaustive key searches (at *64-bits*) [8], [9].

Rijndael encryption comprises of an initial round key addition, followed by a round function (***Nr − 1***)-times and a final round with a slightly altered round function. The round function consists of the ***SubBytes***, ***ShiftRows*** and ***MixColumns*** steps and the round key addition. The phase of the Mix-Columns is omitted in the final round [5].

AES permits a data length of *128 bit* which can be split into four basic operating blocks. These blocks are regarded as a byte array and structured as a ***4 by 4*** matrix which is known as the state. The cipher starts with an ***AddRoundKey*** phase for both encryption and decryption. The output goes through nine primary rounds before reaching the final round, four transformations are conducted during each of those rounds.

Figure 3 shows the process of AES.

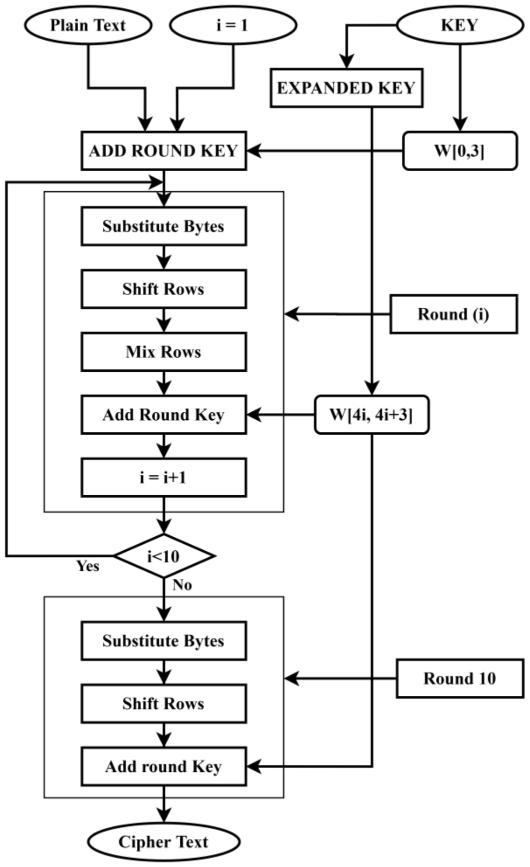
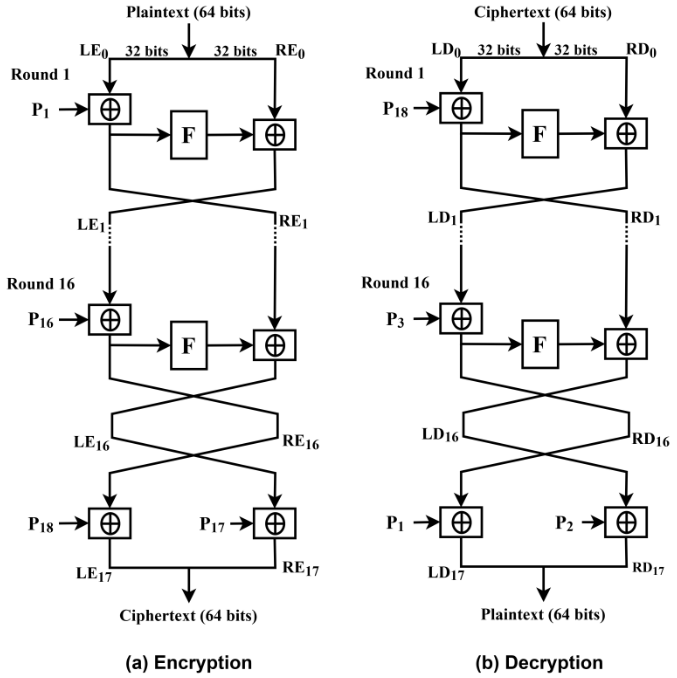


Figure 3: AES Process. Source: [13]

1.4 Blowfish Algorithm

Most of today's encryption algorithms are not available to the public – many of them are patented (e.g. Khufu, REDOC II, and IDEA) or held confidential by governments (e.g., Skipjack and Capstone are protected by the U.S. government). Many of the other algorithms are only partially accessible (such as RC2, RC4, and GOST). The Blowfish algorithm was designed and made publicly accessible by Bruce Schneier, one of the world's leading cryptologists, and the chairman of Counterpane Systems, a consulting company specialized in cryptography and computer security. Blowfish is a *64-bit* block cipher variable-length key. From the very beginning, it was his intention to create this new encryption algorithm to provide a fresh encryption standard for the globe. First introduced in 1993, the Blowfish algorithm has not been cracked yet. It should also be noted that this algorithm can be optimized in hardware apps, although it is often used in software applications, like most other ciphers.

The operation of the blowfish algorithm is in two sections: The key extension section and the encryption section. In the key extension part, a key of length *448 bits* is converted into several sub-key arrays of *4168 bytes* in total. The data encryption takes place via a Feistel network of 16-rounds. Each round is a key-dependent permutation, a key and data-dependent substitution. All computations are XORs and *32-bit* words add-ons. Figure 4 depicts the architecture of the blowfish algorithm.



(a) Encryption (b) Decryption

Figure 4: Blowfish Algorithm. Source: [14]

2. RELATED WORK

Rizvi et al. (2011) studied the encryption speed of Twofish and AES of test files and image files on computers with different sizes of RAM. Their experiment was conducted in a .NET environment using C#, on Windows XP. The authors concluded that Twofish increases in speed as the size of RAM increases [15].

Mandal et al. (2012) compared the implementation and simulation time of DES and AES using MATLAB. They compared the avalanche effect, memory required for implementation and simulation time of DES and AES. The authors observed that AES was the faster algorithm and the memory usage of DES is high whilst AES had a very high avalanche effect [13].

Ramesh and Suruliandi (2013) conducted a performance analysis of DES, AES and Blowfish using metrics such as execution time, memory required and throughput. The authors concluded that Blowfish is about four times faster than AES and twice slower than DES. Blowfish uses less memory when compared with DES and AES. Compared to other algorithms, AES showed poor performance results as it takes more processing power [16].

Kansal and Mittal (2014) evaluated the performance of DES, 3DES and AES on different data types (Text files and Images). The authors observed that AES encryption/decryption time is less than the other algorithms and 3DES took more time for encryption/decryption as it applies the DES algorithm three times [17].

Raigoza and Jituri (2016) compared the performance of AES and Blowfish Algorithms on different types of data strings. Their experimentation was conducted on a Raspberry Pi setup. With respect to speed, the authors concluded that AES is Faster than Blowfish [18].

Panda (2016) conducted a performance evaluation of symmetric (AES, DES, Blowfish) and asymmetric (RSA) encryption algorithms on different types of files such as binary, text and image files. The author concluded that AES has better performance than the other algorithms in terms of both throughput and encryption/decryption time [19].

3. DATA CLASSIFICATION

Data classification enables organizations to think about information based on sensitivity and impact arising from breach of data security, which then enables the organization to evaluate and assess risks associated with the various kinds of data. Reputable standards organizations, such as the International Standards Organization (ISO) and the National Institute of Standards and Technology (NIST), suggest data classification systems so that information can be managed and secured more efficiently according to its relative risk and criticality, advising against procedures that treat all data equally [20]–[22]. Each level of information classification should be linked to a baseline set of security controls providing adequate protection against vulnerabilities, threats and hazards commensurate with the specified level of protection [23].

3.1 Classification Framework

The proposed Confidential Data Classification (CDC) framework outlines two simple classes for records in a dataset; sensitive class and non-sensitive class. To reduced computational demand of encryption, only sensitive class of records in a given dataset are encrypted and the non-sensitive class of records are not encrypted. The CDC framework is an adaptation of existing confidential data categorization standards.

3.1.1 Sensitive Class

The sensitive class encompass records in the dataset that are deemed confidential and includes personal information such as identification numbers, banking information like credit card details, medical records, names, addresses, attorney-privileged documents, intellectual property and patented information, information pertaining to organization secrets etc. Any record which unauthorized disclosure may lead to identity theft, loss of money, potential harm or any adverse effect falls under this class.

3.1.2 Non-sensitive Class

Records classified under non-sensitive class are non-confidential records that have very low or no potential impact risk from unauthorized disclosure. Records that fall under this category include press release information, marketing data, announcements, events, operation logs etc.

3.2 Dataset

The datasets used in this study are datasets of records containing human resource data of employee records and was downloaded from http://eforexcel.com (eforexcel, accessed 6.27.19) [24]. The simulation was conducted on 10 datasets containing 1000 to 10000 records. Table 2 shows the number of columns/fields in the datasets and the classification parameters associated with the fields.

Table 1: Classification Parameters

Index	Fields	Classification
0	Emp ID	Sensitive
1	Name Prefix	Sensitive
2	First Name	Sensitive
3	Middle Initial	Non-sensitive
4	Last Name	Sensitive
5	Gender	Non-sensitive
6	E-Mail	Sensitive
7	Father’s Name	Sensitive
8	Mother’s Name	Sensitive
9	Mother’s Maiden Name	Sensitive
10	Date of Birth	Sensitive
11	Time of Birth	Non-sensitive
12	Age in Yrs.	Sensitive
13	Weight in Kgs.	Non-sensitive
14	Date of Joining	Non-sensitive
15	Quarter of Joining	Non-sensitive
16	Half of Joining	Non-sensitive
17	Year of Joining	Non-sensitive
18	Month of Joining	Non-sensitive
19	Month Name of Joining	Non-sensitive
20	Short Month	Non-sensitive
21	Day of Joining	Non-sensitive
22	DOW of Joining	Non-sensitive
23	Short DOW	Non-sensitive
24	Age in Company	Non-sensitive
25	Salary	Sensitive
26	Last % Hike	Non-sensitive
27	SSN	Sensitive
28	Phone No.	Sensitive
29	Place Name	Non-sensitive
30	County	Non-sensitive
31	City	Non-sensitive
32	State	Non-sensitive
33	Zip	Non-sensitive
34	Region	Non-sensitive
35	User Name	Sensitive
36	Password	Sensitive

4. SIMULATION DESIGN

For the simulation, the same platform with the following specifications was used:

- **Processor**: Intel® Core™ i7-7600U @2.80GHz (4 CPUs), ~2.9GHz
- **Memory**: 16 GB of RAM
- **Operating System**: Windows 10 Enterprise 64-bit (10.0, Build 18362)

In the implementation of RSA encryption, Optimal Asymmetric Encryption Padding (OAEP) is used as a padding scheme for RSA encryption since the traditional implementation of RSA is insecure [25]. The purpose of using OAEP for padding is to: add a random element that can be used to transform a deterministic system of encryption (traditional RSA) into a probabilistic system; and prevent the leakage of information or partial decryption of ciphertexts by ensuring that an attacker cannot recover any portion of the plaintext without inverting the one-way trapdoor permutation [26]. OAEP is standardized in PKCS#1, foremost of the Public-Key Cryptography Standards family [27], [28].

In the implementation of the symmetric encryption algorithms (3DES, AES, Blowfish), the block cipher mode of operation adopted is the Cipher Feedback (CFB) mode. CFB mode was adopted to address issues encountered with other modes that required special measures in other to process data with different lengths from multiples of the block cipher which has a fixed size (block size). With CFB mode, padding of the block cipher is not necessary which is an advantage because the block cipher is only used during encryption and the data or message being encrypted does not need to be padded to a multiple of the block size [29]–[31].

4.1 Time and Space Complexity

Performance comparison analysis is largely based on results obtained from the python profiler libraries: CProfile and Pstats which were implemented to capture the runtime and memory usage of the classification framework and encryption algorithms. The assumptions made was that the runtime and memory usage reported by the python profiling libraries is the time complexity and space complexity of the classification framework with the encryption algorithms.

4.2 Observations

In observing the simulation process, some factors which may, to some extent, affected the performance of the encryption process were noticed and are stated below:

- The information in some records has shorter/longer length than others, we believe this contributes to the non-linearity of the results and consequently the line graph describing the execution time, as this affects the encryption time for some records in the datasets.
- Unpreventable windows background processes (operating system processes needed for necessary functions and performance) may have affected the accuracy of the execution time for some of the simulations performed.
- Occasional, mild usage of some application such as word processor and browser on the simulation platform while the simulations were running may have affected the accuracy of the execution time for some of the simulations conducted.

5. RESULTS

Figure 5 below illustrates the processing time of complete encryption and classification with encryption for all the chosen encryption algorithms. RSA takes the longest time to finish and therefore the slowest. Blowfish is the fastest algorithm followed by AES and 3DES.

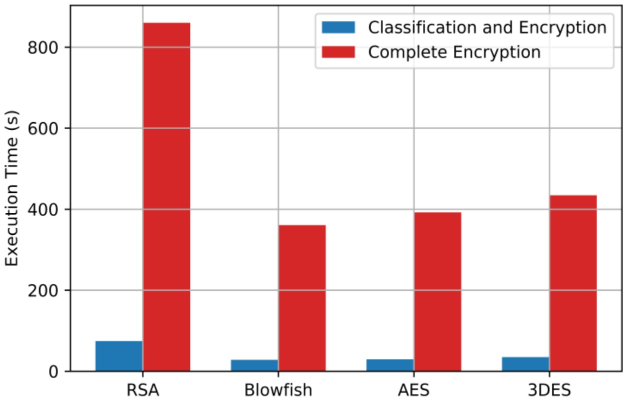


Figure 5: Comparison of Execution Time of Classification with Encryption Techniques

Figure 6 below illustrates the memory usage of complete encryption and classification with encryption for all the chosen encryption algorithms. The memory usage of RSA is the highest. Blowfish, AES and 3DES use the same amount of memory for classification with encryption and complete encryption.

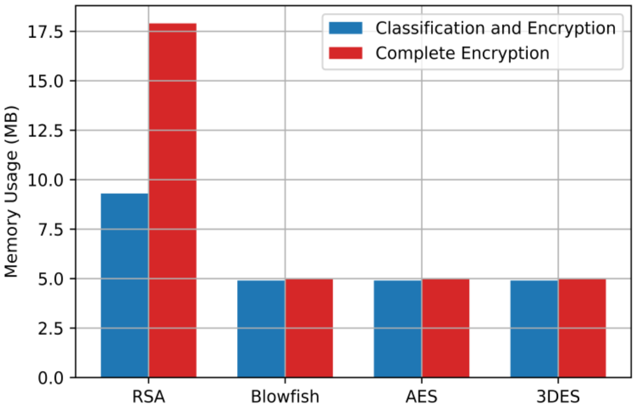


Figure 6: Comparison of Memory Usage of Classification with Encryption Techniques

6. CONCLUSION

This work evaluates the performance of AES, 3DES, Blowfish and RSA on records in a given dataset using two metrics: computing time and memory usage. In terms of processing time, simulations conducted shows that Blowfish is faster than AES which is, in turn, faster than 3DES and RSA was the slowest algorithm. For memory usage, the symmetric (AES, 3DES, Blowfish) encryption algorithms used the same amount of memory but the asymmetric (RSA) encryption algorithm used about twice the amount of memory used by the symmetric algorithms.

7. REFERENCES

[1] V. Kumar, S. Chaisiri, R. Ko, and Institution of Engineering and Technology, Data security in cloud computing. 2017.

[2] R. Arora, A. Parashar, and C. C. I. Transforming, “Secure user data in cloud computing using encryption algorithms,” International journal of engineering research and applications, vol. 3, no. 4, pp. 1922–1926, 2013.

[3] J.-P. Aumasson and M. D. Green, Serious cryptography: a practical introduction to modern encryption. San Francisco: No Starch Press, 2017.

[4] J. R. Vacca, Ed., Cloud computing security: foundations and challenges. Boca Raton: CRC Press/Taylor & Francis Group, 2017.

[5] H. Delfs and H. Knebl, Introduction to Cryptography Principles and Applications. Springer, 2015.

[6] J. Buchmann, Introduction to cryptography. Springer Science & Business Media, 2013.

[7] W. Stallings, Cryptography and network security: principles and practice. Pearson Upper Saddle River, 2017.

[8] G. Singh, "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security," International Journal of Computer Applications, vol. 67, no. 19, 2013, doi: 10/gf3359.

[9] A. Nadeem and M. Y. Javed, "A performance comparison of data encryption algorithms," in 2005 international conference on information and communication technologies, 2005, pp. 84–89, doi: 10/b8t8p6.

[10] D. E. Standard, "Federal information processing standards publication 46," National Bureau of Standards, US Department of Commerce, vol. 23, 1977.

[11] J. Nechvatal et al., "Report on the development of the Advanced Encryption Standard (AES)," Journal of Research of the National Institute of Standards and Technology, vol. 106, no. 3, p. 511, 2001, doi: 10/gf4wxq.

[12] T. Jamil, "The rijndael algorithm," IEEE potentials, vol. 23, no. 2, pp. 36–38, 2004, doi: 10/cx33c3.

[13] A. K. Mandal, C. Parakash, and A. Tiwari, "Performance evaluation of cryptographic algorithms: DES and AES," in 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, 2012, pp. 1–5, doi: 10/gf4zf9.

[14] S. O. M. Kamel, M. S. El Sherif, A. S. T. El Dein, and S. A. El Rahman, "Novel TEA Algorithm for IP Telephony System," International Journal of Informatics and Communication Technology (IJ-ICT), vol. 1, no. 1, pp. 6–19, 2012, doi: 10/gf4zpp.

[15] S. A. M. Rizvi, S. Z. Hussain, and N. Wadhwa, "Performance Analysis of AES and TwoFish Encryption Schemes," in 2011 International Conference on Communication Systems and Network Technologies, Katra, Jammu, India, 2011, pp. 76–79, doi: 10/fkpd2j.

[16] A. Ramesh and A. Suruliandi, "Performance analysis of encryption algorithms for Information Security," in 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, 2013, pp. 840–844, doi: 10/gf9rhq.

[17] S. Kansal and M. Mittal, "Performance evaluation of various symmetric encryption algorithms," in 2014 International Conference on Parallel, Distributed and Grid Computing, 2014, pp. 105–109, doi: 10/gf4wx8.

[18] J. Raigoza and K. Jituri, "Evaluating Performance of Symmetric Encryption Algorithms," in 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2016, pp. 1378–1379, doi: 10.1109/CSCI.2016.0258.

[19] M. Panda, "Performance analysis of encryption algorithms for security," in 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, Odisha, India, 2016, pp. 278–284, doi: 10.1109/SCOPES.2016.7955835.

[20] G. Disterer, "ISO/IEC 27000, 27001 and 27002 for information security management," 2013, doi: 10/gfvhsf.

[21] F. Pub, "Standards for Security Categorization of Federal Information and Information Systems," NIST FIPS–199, 2004.

[22] S. Zevin, Standards for security categorization of federal information and information systems. DIANE Publishing, 2009.

[23] S. Radack, "Managing Information Security Risk: Organization, Mission and Information System View," National Institute of Standards and Technology, 2011.

[24] Eforexcel, "Sample CSV Files / Data Sets for Testing - Human Resources," E for Excel | Awakening Microsoft Excel Student Inside You. [Online]. Available: http://eforexcel.com/wp/downloads-16-sample-csv-files-data-sets-for-testing/. [Accessed: 27-Jun-2019].

[25] M. Bellare and P. Rogaway, "Optimal asymmetric encryption," in Workshop on the Theory and Application of of Cryptographic Techniques, 1994, pp. 92–111.

[26] V. Shoup, "OAEP reconsidered," in Annual International Cryptology Conference, 2001, pp. 239–259, doi: 10/fj9f2f.

[27] J. Jonsson and B. Kaliski, "Public-key cryptography standards (PKCS)# 1: RSA cryptography specifications version 2.1," 2003, doi: 10/btnp.

[28] D. H. Phan and D. Pointcheval, "OAEP 3-round: A generic and secure asymmetric encryption padding," in International Conference on the Theory and Application of Cryptology and Information Security, 2004, pp. 63–77.

[29] M. Bellare and P. W. Rogaway, "Block cipher mode of operation for secure, length-preserving encryption," Sep-1997.

[30] M. Dworkin, "Recommendation for block cipher modes of operation: methods for format-preserving encryption," NIST Special Publication, vol. 800, p. 38G, 2016.

[31] M. Dworkin, "Recommendation for block cipher modes of operation. methods and techniques," National Inst of Standards and Technology Gaithersburg MD Computer security Div, 2001.