International Journal of Information Engineering and Applications
2018; 1(2): 61-66
http://www.aascit.org/journal/information

American Association for Science and Technology

Application of SHA-256 in Formulation of Digital Signatures of RSA and Elgamal Cryptosystems

Aderemi Elisha Okeyinka[*], Oluwatobi Alao, Babatunde Gbadamosi, Roseline Oluwaseun Ogundokun

Department of Computer Science, Landmark University, Omu-Aran, Nigeria

Email address

okeyinka.aderemi@lmu.edu.ng (A. E. Okeyinka)
[*]Corresponding author

Citation

Aderemi Elisha Okeyinka, Oluwatobi Alao, Babatunde Gbadamosi, Roseline Oluwaseun Ogundokun. Application of SHA-256 in Formulation of Digital Signatures of RSA and Elgamal Cryptosystems. *International Journal of Information Engineering and Applications.* Vol. 1, No. 2, 2018, pp. 61-66.

Received: January 25, 2018; **Accepted**: February 18, 2018; **Published**: March 23, 2018

Abstract: In this study, the two cryptosystems are considered for computational speed efficiency using SHA-256 in formulation of digital signature. The goal of this study is to observe whether or not the application of SHA-256 hash function would yield any significant difference in the literature position that RSA is a more efficient cryptosystem than Elgamal. The methodology employed involves Java programming implementation of the RSA and Elgamal cryptosystems using SHA-256 hash function. Ten different text data of various sizes are used as input data, and the internal clock of the computer is used in monitoring and calculating the computational speeds of both the RSA and Elgamal. The results obtained show that there is no significant difference in the computational speeds of both strategies. This is unlike the earlier results found in the literature which shows that RSA is more efficient. A plausible explanation for this is that the SHA-256 hash function has a significant effect on the implementation. However, since SHA-256 is just one of the versions of SHA-2 hash family, a study that will consider other members of the SHA-2 family is recommended to enable us formulate a theorem or conjecture on the effect of hash function on cryptosystems, and in particular the RSA and Elgamal cryptosystems.

Keywords: Cryptosystems, Digital Signature, Elgamal, RSA, SHA-256

1. Introduction

Issues in information security and efficiency of algorithms are of increasing practical importance to computer scientists, and digital communication engineers [1] [6]. Messages are being exchanged over worldwide publicly accessible computer networks [2], and such messages are designed with tools that can provide confidentiality, authentication, data integrity, and non-repudiation. RSA and Elgamal are two of the existing cryptographic algorithms. RSA is a public key, it is a bijective function and computationally efficient. It was developed by Rivest, Shamir, and Adleman. On the other hand, Elgamal is a discrete logarithm. It is a one-way function, and contains no trap door. The computational complexity study of RSA and Elgamal has been carried out [7] [8], yielding a general finding that RSA is more efficient than Elgamal. Some of the other cryptographic algorithms in use are: Data encryption standard (DES), Hash also known as Fingerprint or Message Digest and MD5.

1.1. Secure Hash Algorithm

Secure Hash Algorithm (SHA) developed by the National Institute of Standards and Technology (NIST) was also designed on the same principle as MD4 and was published as Federal Information Processing Standard (FIPS 180) in 1993. A revised version was issued as FIPS180-1 in 1995 and is generally referred to as SHA-1. When revised version of SHA-1 was published no details of the weaknesses found in SHA-0 (originally SHA) were provided. SHA-1 produces a hash value of 160 bit. In 2002, NIST produced a revised version of the standard known as FIPS180-2 and defined three new versions of SHA with digest lengths of 256, 384 and 512 and known as SHA-256, SHA-384, and SHA-512 respectively [3]. So total SHA versions become four including SHA-1 (160 bit). In October 2008, FIPS 180-2 has been replaced by FIPS 180-3 and in new standard SHA-224 has been added which is same as other SHA algorithm producing 224 bits of the message digest. All these SHA versions are based on the same principle of MD4 and hash length has changed and certain other improvements have been carried from one version to next. Attacks on SHA-0 and SHA-1 have been reported in various research works [10].

1.2. Digital Signature Schemes and Hash Functions

One of the problems with the classical digital signature schemes is that the signatures are as long as or longer than the messages that they sign. When the messages are large this can become a significant difficulty. One way to deal with this is to use cryptographic hash functions.[9] A hash function h takes a message m of arbitrary length and produces a message digest h(m) of some fixed length. In order for a hash function to be useful in cryptographic work, it should satisfy the following conditions:

a. The message digest h(m) should be calculated very quickly.
b. The hash function h should be a one-way function, that is, given a message digest h(m), it should be computationally infeasible to obtain the message m.
c. The hash function h should be strongly collision-free, meaning that it should be computationally infeasible to find two messages m1 and m2 so that h(m1) = h(m2).

2. Research Motivation

The fundamental reason for analyzing complexity of algorithms is to know which of the several algorithms solving the same problem is best for practical purposes. Furthermore, there is the need to establish under what circumstances a given algorithm performs better than the other. For example, quick sort technique is generally speaking superior to bubblesort strategy. But, for a small size data file bubblesort performs better than quicksort. Moreover, it has been found that RSA is more energy efficient than Elgamal [4]; also it has been established that RSA performs better in terms of computational speed [7]. However, other performance parameters need to be studied and applied on these two algorithms in order to be able to formulate a theorem that can capture the behaviour of the algorithms. So the rationale for this study is to determine and compare the computational speeds of RSA and Elgamal using the Hash Function, and in particular SHA-256. RSA was proposed in 1977 [4], while Elgamal was proposed in 1985 [4]. RSA is a deterministic algorithm and appears to be the most popularly implemented public-key cryptosystem [5]. Elgamal on the other hand is a non-deterministic algorithm [5], and an extension of the Diffie-Hellman key agreement protocol. Hence the goal of this study is to investigate one deterministic and one non-deterministic algorithms for computational speed analysis using SHA-256.

3. Method

The RSA and Elgamal cryptosystems are implemented using java in the same programing environment. The SHA-256 hash function is used to generate a fixed unique value, also called Message Digest (MD) for a given message M of arbitrary length. The MD is then encrypted to generate the signature for the message. Each cryptosystem consists of four phases viz: key generation, MD calculation, encryption and decryption, and, signing and verification. The java platform takes ten different text data one at a time, as input; each character of the text is converted into its ASCII value and then used as appropriate in computing cipher text information. The cipher text information, when decrypted results in the original message. The length of the input text is determined by the java program. The computers' internal clock is used to determine the execution time of the input texts. Hence a comparative analysis is done to know which of the RSA and Elgamal cryptosystem is computationally more efficient. The following algorithms are used:

a. Key Generation Algorithm
b. Encryption and Decryption Algorithm
c. Signing and Verification Algorithm

All of these algorithms and SHA-256 are implemented in the same programming environment using JAVA.

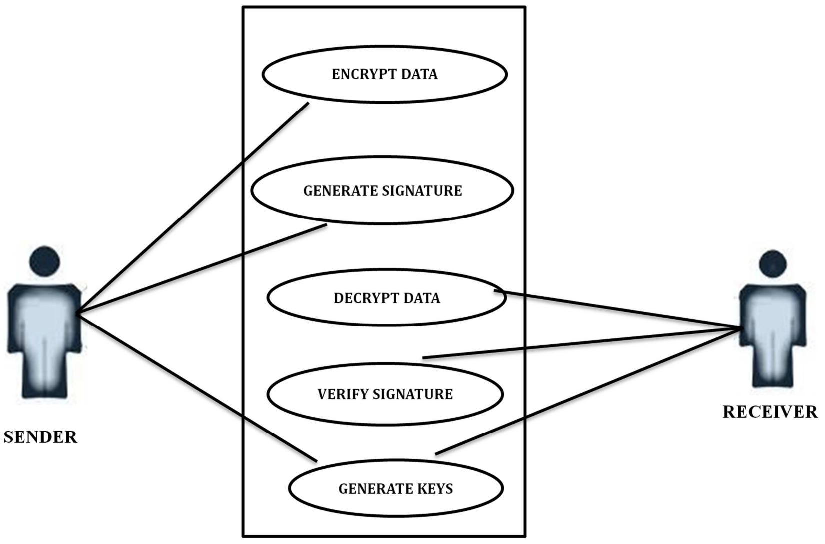

Figure 1. *Use Case Diagram: This figure shows the activities carried out by the traditional Alice and Bob, referred here as Sender and Receiver.*

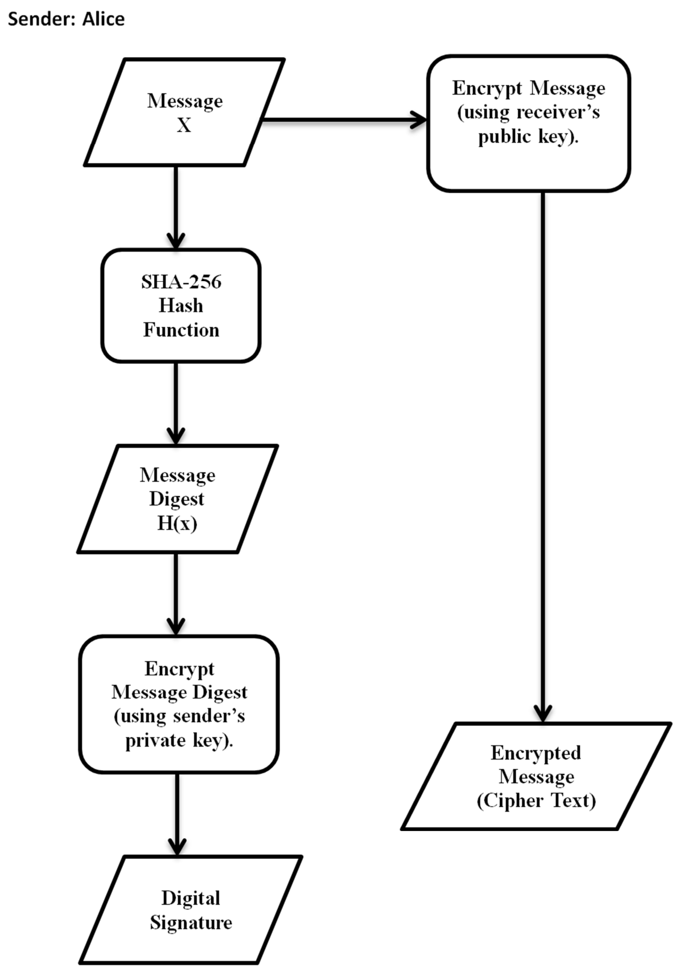
Sender: Alice
Message X
Encrypt Message (using receiver's public key).
SHA-256 Hash Function
Message Digest H(x)
Encrypt Message Digest (using sender's private key).
Encrypted Message (Cipher Text)
Digital Signature

Figure 2. *Encryption and Digital signature from sender.*

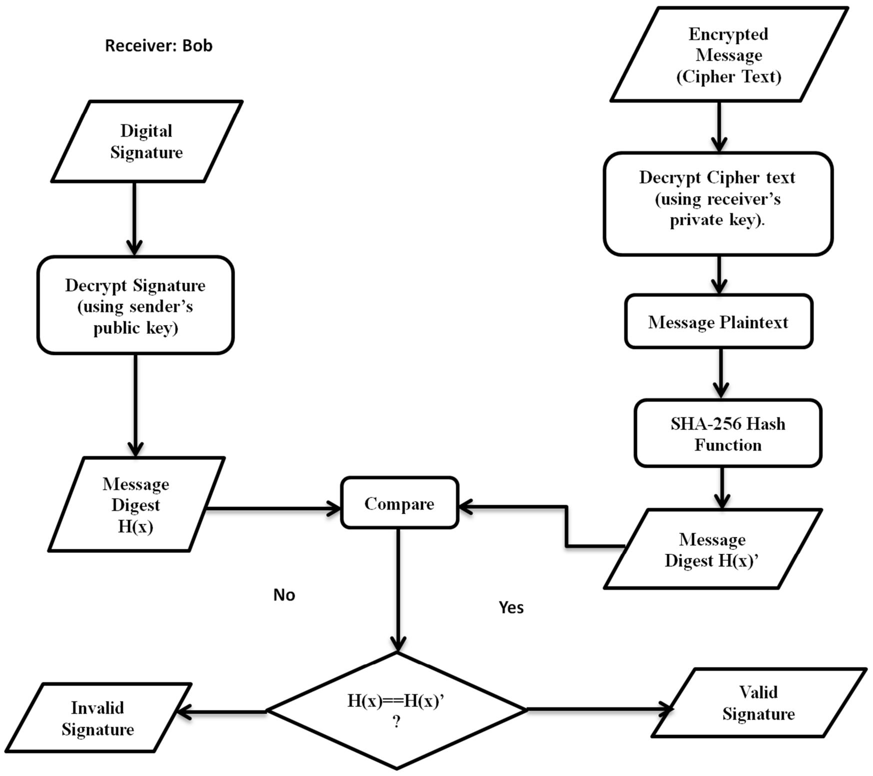

Figure 3. *Decryption and Message Digest Verification.*

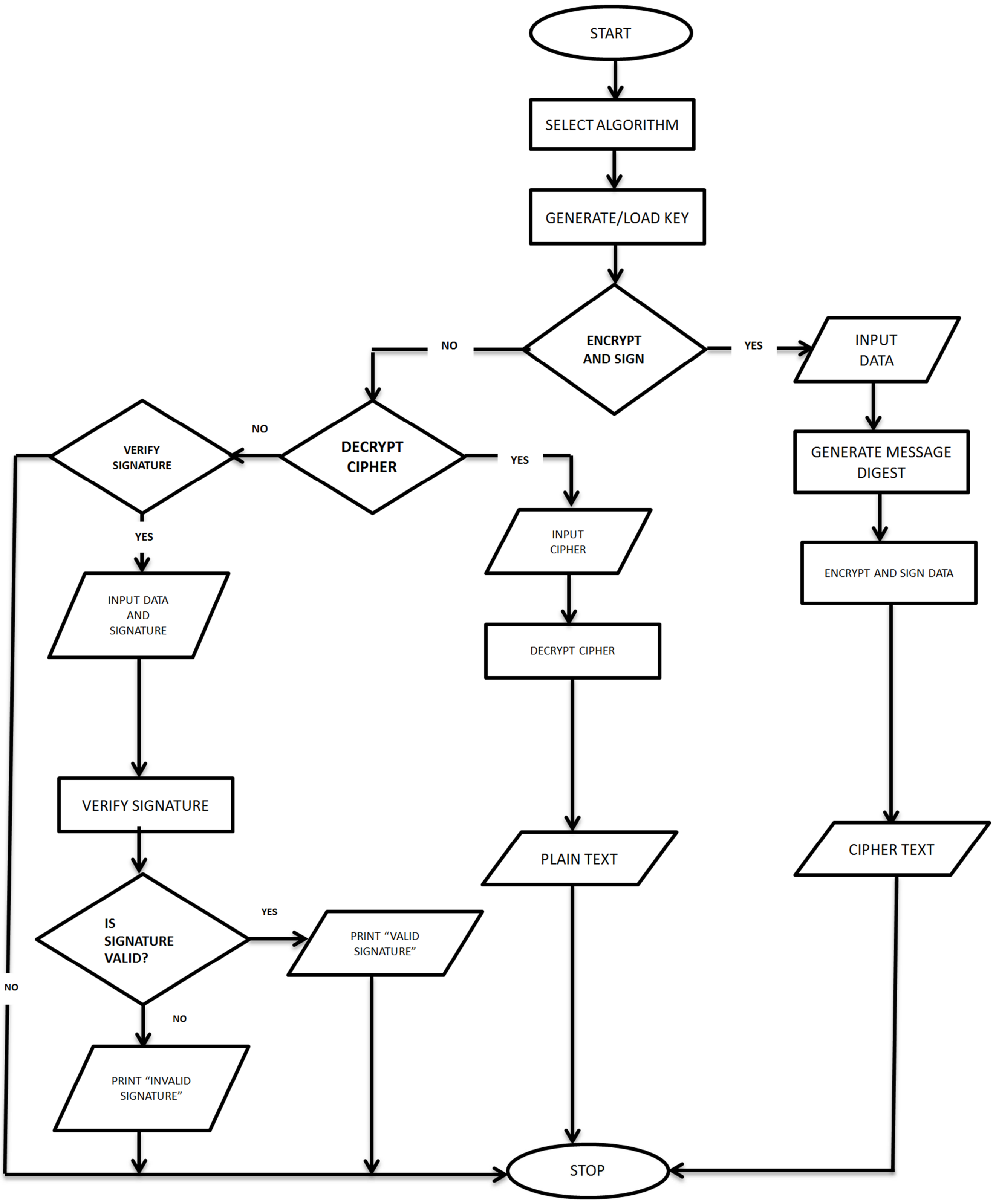

Figure 4. *User Interface Design.*

4. Implementation

The following constitute the main modules in the Java system developed.

i. Encrypting of files using RSA and Elgamal algorithms
ii. Signature generation and verification
iii. Decrypting of information using the RSA and Elgamal algorithms
iv. GUI interface for easy interaction
v. Auto-generation of private and public keys for

encryption, signing and decryption.

vi. Digital signature schemes and hash functions

vii. Interface for selection of file or document to be signed or encrypted.

The user interface (figure 4) is the abstraction of the system behaviour.

In figure 2 the sender applies SHA-256 to produce the digital signature; the receiver also re-computes the digital signature (figure 3) for validity. If the two signatures from the sender and the receiver are equal, then the integrity of the information is ascertained otherwise the information has been altered in the process of movement/communication.

5. Results

The following are the results obtained when the algorithms used were implemented on ten different text data.

Table 1. *Encryption.*

	Length Of Characters	Size (KB)	Time Taken (RSA) (ms)	Time Taken (Elgamal) (ms)
1	1078	2	18	288
2	2224	3	40	604
3	3379	4	38	956
4	4515	5	47	1311
5	5730	6	65	2034
6	6963	7	76	2372
7	8132	9	106	2761
8	9376	10	94	3215
9	11720	12	136	4359
10	13761	14	229	4689

Table 2. *Decryption.*

	Length Of Characters	Size (KB)	Time Taken (RSA) (ms)	Time Taken (Elgamal) (ms)
1	1078	2	2228	336
2	2224	3	4707	676
3	3379	4	6909	1031
4	4515	5	9400	1405
5	5730	6	11681	2149
6	6963	7	14123	2697
7	8132	9	16577	2999
8	9376	10	18984	3526
9	11720	12	23477	4321
10	13761	14	27609	5617

Table 3. *Signature Generation.*

	Length Of Characters	Size (KB)	Time Taken (RSA) (ms)	Time Taken (Elgamal) (ms)
1	1078	2	285	36
2	2224	3	269	39
3	3379	4	284	45
4	4515	5	293	38
5	5730	6	264	47
6	6963	7	273	34
7	8132	9	286	36
8	9376	10	293	46
9	11720	12	296	31
10	13761	14	281	36

Table 4. *Signature Verification.*

	Length Of Characters	Size (KB)	Time Taken (RSA) (ms)	Time Taken (Elgamal) (ms)
1	1078	2	3	73
2	2224	3	3	79
3	3379	4	2	79
4	4515	5	3	84
5	5730	6	3	65
6	6963	7	3	67
7	8132	9	4	82
8	9376	10	3	80
9	11720	12	3	79
10	13761	14	2	66

6. Discussion

The computational speeds/times taken for the execution of RSA and Elgamal algorithms are presented on the Tables. The results show that RSA performs better than the Elgamal in encryption and signature verification, while Elgamal performs better in decryption and signature generation. The use of hash functions provides assurance to the receiver about the integrity of the information received. The integrity check helps the receiver to detect any changes made to the original file. These results are not sufficient to make a general statement about the holistic behaviour of these two cryptosystems. Unlike what already exists in literature where RSA performs better in most of the algorithms than the Elgamal, the application of the SHA-256 hash function appears to affect the present results significantly, in that there is no obvious superiority of one cryptosystem over the other in the results obtained.

7. Conclusion

The family of SHA comprises SHA-0, SHA-1, SHA-2 and SHA-3 [10]. The first member of the family, that is, SHA-0, was published in 1993. Between 1993 and now, many other versions of the SHA have been developed arising from weaknesses discovered in the earlier versions [5]. SHA-256 which is used in this study belongs to SHA-2 family. Other variants of SHA-2 are SHA-224, SHA-384 and SHA-512 depending upon the number of bits in their hash values. In order for us to know the effectiveness of SHA in the RSA and Elgamal cryptosystems, it is considered necessary that more variants of SHA should be applied to the cryptosystems. A comparative study of various SHA can then be carried out to have a better informed effect of application of hash functions in cryptosystems.

References

[1] Alfred J. Menezes, Vanstone, S. A., & van Oorschot, P. C. (1997). *Handbook of applied cryptography*. CRC press.

[2] Elgamal T. (1985), A public key cryptosystems and signature scheme based on discrete logarithms. IEEE trans Inf Theory 31 (4): 469-472.

[3] Fips, P. U. B. (2000). 186-2. Digital signature standard (DSS). *National Institute of Standards and Technology (NIST)*.

[4] Hans Deifs and Helmut Knebl (2006), Introduction to cryptography, principles and applications: second Edition, Springer publisher.

[5] International Association of Engineers [online]. Available: http://www.iaeng.org (Elbirt AJ (2008). Understnding and applying cryptography and data security. Auebach publications, Taylor and Francis Group).

[6] Kessler, G. C. (2016). An Overview of Cryptography (Updated Version, 3 March 2016).

[7] Okeyinka A. E. (2015), computational speeds analysis of RSA and Elgamal algorithms on text data. In proceedings of the world congress on engineering and computer science 2015, WCECS 2015, Lecture Notes in Engineering and Computer Science, 21-23 October 2015, San Fransisco, USA, pp 115-118.

[8] Okeyinka A. E. (2017), Computational Complexity Study of RSA and Elgamal Algorithms; Transactions on Engineering Technologies; Springer Nature Singaporw Pte Ltd. DOI 10.1007/978-981-10-2717-8_17.

[9] Rednam Venkata Ramana, B. Lakshmana Rao, (2013)," Design and Implementation of Digital Signatures", *International Journal of Latest Trends in Engineering and Technology (IJLTET).*

[10] Sobti, R., & Geetha, G. (2012). Cryptographic hash functions: a review. *IJCSI International Journal of Computer Science Issues*, *9* (2), 461-479.