ANALYSIS AND PERFORMANCE EVALUATION OF SELECTED DIGITAL SIGNATURE ALGORITHMS

O. EFEVBERHA-OGODO[1] and P. AIGBE[2]

[1,2]Department of Mathematics and Computer Science, Western Delta University, Oghara, Delta State
Corresponding Author: Email- efevbaire@gmail.com, Phone- +2348061295250

Abstract
The advancement of information and communications technology, especially the internet, has created a need to continuously seek optimal solutions to the issue of security. Digital signature is by far one of the most important cryptographic techniques used in e-government and e-commerce applications to enable communication security. It provides authentication of senders and receivers of online messages and transactions, and offers non-repudiation of messages. This paper presents a study into the analysis and performance evaluation of the DSA, RSA ElGamal and Schnorr algorithms based on key performance indicators. A number of scholarly works about digital signature research were considered and reviewed. The formulae for generating each Signature Key, Signing and Verification functions were coded in C programming language. Each digital signature considered in this study was evaluated based on their performance when executed in a computer system. The selected algorithms are effective in securing data and information. DSA has an advantage over others in terms of Signature Generation and RSA is most suitable for situations where timing is critical.

Keywords: Digital Signature Algorithm, Key Generation, Signature Generation, Public Key, Private Key

INTRODUCTION

Technological advancement may have been empowering to man as access to all kinds of information is made available but it comes at a price. The volume of data that has to be dealt with especially in a wide area network such as the Internet can be overwhelming and even difficult to manage (Etzel, 1995 and Helmy *et al*. 2003). It can really be burdensome to acquire and recall relevant information on the World Wide Web (WWW). It is therefore important that sensitive information is passed and not lost or tampered with in midst of the abundance of data in the cloud.

In communication and networks, digital signatures are essential in verifying messages sent across various platforms whereby the receiver is guaranteed of its source. There are two major characteristics of digital signatures that make them favourable: authentication and non-repudiation. It allows a particular receiver to detect any forgery attempts on a message and also a sender cannot deny sending when a digital signature has verified the message. A digital signature can be described as a mathematical concept especially useful in financial transactions and cases where it is important to detect forgery (Panjwani & Mehta, 2015).

A digital signature scheme is basically categorized into 3 parts; key generation algorithm, signing algorithm and signature verifying algorithm. At the sender's end, public and private keys are generated then a signature is digitally signed before sending to authenticate its origin. While at the receiver's end, the encrypted message is received and the signature on the message is subjected to verification using the sender's public key.

A Digital Signature Standard (DSS) was stated decades ago by the National Institute of Standards and Technology (NIST). It enumerates algorithms that can be used to generate digital signatures including a hash function referred to as Secured Hash Algorithm (SHA-1). (Gilani and Mir, 2009)

A simple digital signature algorithm will follow the procedure depicted in Figure 1.

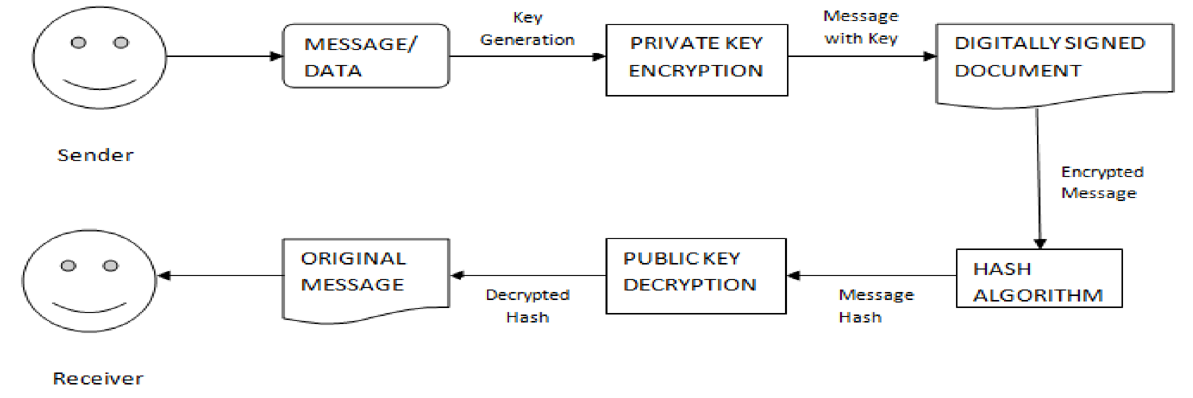

Fig. 1: Block diagram showing the processing of a Digital Signature.

At the sender's end:

i. Public and private keys are generated.
ii. Sender's signature is appended to the message and message is sent.

At the Receiver's end:

i. The encrypted message is received.
ii. Message digest is verified by sender's public key.
iii. The resulting value must be equal to the signed message by the Sender. In this way the Receiver gets confirmed about the authenticity of the Sender.

Digital Signature Schemes for Data Encryption and Decryption

This section explores the digital signature algorithms selected for this study, their various steps taken to generate signature and attaching it to a message.

Digital Signature Algorithm (DSA)

There are different types of algorithms but the most common is the Digital Signature Algorithm (DSA) for which variants were developed in an attempt to improve its features. An example is the ECDSA. To sign a message M, the sender generates a random key and signs the messages while the recipient carries out signature verification. The following parameters can be found in a DSA:

Key Generation

A value for x is chosen by a random method, where $0 < x < q$

The value $y = g^x \bmod p$ (1)

Public key is (p, q, g, y). Private key is x

Signature Signing

A random value k is generated for each message, where 0< k < q

$r = (g^{k} \bmod p) \bmod q$ (2)

$s = k^{-1} (H(m) + xr) \bmod q$ (3)

The signature is (r, s). If r or s is equal to zero, then generate random value k

Signature Verification

Signature is valid if (0 < r < q) or (0 < s < q) is satisfied.

The value $w = s^{-1} \bmod q$ (4)

The value $u1 = [H(M) w] \bmod q$ (5)

The value $u2 = (rw) \bmod q$ (6)

The value $v = [(g\ u1.\ y\ u2) \bmod p] \bmod q$ (7)

If v = r then the signature is verified

(Chu et al, 2003)

RSA Digital Signature Algorithm

Another commonly used digital signature is the RSA scheme developed by Ron Rivest, Adi Shamir and Len Adleman and consequently named after each of them. Due to its top security measures and simple implementation, the RSA cryptographic algorithm has gained widespread popularity. However, there may be a few bottlenecks when computing the modular exponentiation of very huge numbers (Xiao et al, 2015).

RSA Key Generation

The following steps are required to generate a key needed to encrypt data/information in an RSA scheme.

i. Two random primes, p and q, are chosen
ii. Evaluate n = pq (8)
iii. φ (phi) = (p-1) (q-1) (9)
iv. choose e, such that gcd(e, phi) = 1
v. compute d such that 1 < d < phi
vi. ed is equivalent to 1(mod phi)
vii. public key (e, n), private key (d)

RSA Signature Generation

Here, the generated key is used to compute the digital signature.

i. compute $s = m^{d} \bmod n$ (10)
ii. signature is s

RSA Signature Verification

At the receiver’s end, the original message m is gotten by decrypting the sent message using the public key (e, n) and the signature s.

i. compute $m = s^{e} \bmod n$ (11)

ElGamal Digital Signature Algorithm

In 1985, Taher ElGamal proposed a digital signature scheme which depends on the security of a private key x. If and when the private key is intercepted by an intruder, posing danger to the message sent, then a secure transmission will no longer exists causing anyone to have access. This may require the value of the private key and/or public key to be changed from time to time by the signer (Xiao-fei et al, 2010).

ElGamal Key Generation

The steps required to generate an encryption key in the ElGamal scheme includes the following:

i. The private key x is randomly chosen, where $1 < x < p-1$
ii. The public key $y = g^{x} \bmod p$ (12)

ElGamal Signature Generation

The steps involved in ElGamal signature generation include:

i. A random integer k per message is chosen such that $1 < k < q-1$
ii. $\gcd(k, q-1) = 1$ (13)
iii. The digital signature $m = (r, s)$
iv. The value $r = g^{k} \bmod p$ (14)
v. The value $s = [\,H(m) - x.r) \,.\, k^{-1} \bmod (p-1)]$ (15)

ElGamal Signature Verification

The receiver of the encrypted message can verify its sender by implementing the following steps.

i. $M(r, s)$ is correct if $1 \leq r \leq p$
ii. $y^{r} .r^{s}$ is equivalent to $g^{H(m)}$ (mod p)

Schnorr Digital Signature Algorithm

Schnorr digital signature parameters include:

Key Generation

Choose suitable primes p and q

Choose g such that $g^{q} = 1 \bmod p$

A private key, x (number) is generated where $0 < x < q$

A public key, $y = g^{x} \bmod q$ (16)

Signature Generation

A random value k is chosen such that $0 < k < q$

Compute $r = g^{k} \bmod p$ (16)

Concatenate message with x plus hash using $e = H(M \,||\, r)$ (17)

Compute $s = (k + xe) \bmod q$ (18)

Signature pair is (s, e)

Signature Verification

$m(r, s)$ is true if $H(m \,|\,| g^{s}. y^{-r} \bmod p) = r$

Performance Analysis of Selected Digital Signature Schemes

The “goodness” of an algorithm can be appraised by a variety of performance criteria. One of the key factors to be considered is time, that is, speed taken in the algorithm execution. There are several aspects of such time criterion. One might be concerned with the execution time required by different algorithms for solution of a particular problem on a particular computer system. However, such an empirical measure is strongly dependent upon both the program and the machine used to execute the algorithm. Furthermore, if the algorithms are compared on computers with varying system specifications, slightly different conclusions may be reached. The second factor considered in this study is the amount of space or computer memory required by each algorithm, which will be measured in bytes.

RESEARCH METHODOLOGY

Digital Signatures are one of the most important cryptographic techniques used in e-government and e-commerce applications to enable communication security. This study takes a quantitative research approach carried out on the selected signature schemes in order to sufficiently understand their processes. Random integers were used to calculate and implement the varying equations for each of the algorithms earlier stated.

The C programming language was used to implement each the algorithm requirements due to its inherent characteristic of combining the features of both low-level and high-level languages. The program codes were executed using different data sizes in bytes and the time taken to execute each data size was recorded in milliseconds. The results were analysed and evaluations were made about the performance of the selected digital signature schemes.

Systems Configuration for Algorithm Analysis

The following provides a brief summary of the hardware/and software configuration of the computer system used to perform comparative analysis.

- Manufacturer: Hewlett-Packard
- Model: HP Compaq67356s
- Rating: AMD Sempron (tm) SI-40 2.00GHz
- RAM: 2.00 GB
- Hard Disk: 150 GB
- System Type: 32 bit Operating System
- Operating System: Windows Vista

RESULTS

The results of the time taken were recorded in milliseconds after each algorithm was executed with varying data sizes in bytes. The table below displays a distinction from each algorithm.

Table 1: Signature Generation Time based on data size

S/N	Data size in Bytes	Digital Signature Algorithms Performance time in Milliseconds (ms)			
		DSA	RSA	ElGamal	Schnorr
1	10	4.78	5.22	10.55	10.27
2	20	5.87	6.26	11.37	12.30
3	30	6.75	7.26	13.73	14.55
4	40	7.58	9.84	19.55	14.72

For Table 1, data measured in bytes was plotted on the x-axis while time measured in milliseconds can be seen on the y-axis. Random values were chosen and executed with other variables in stated in each of the algorithm procedures for generating a signature on a message. The result of the plotted values are shown in the graph below.

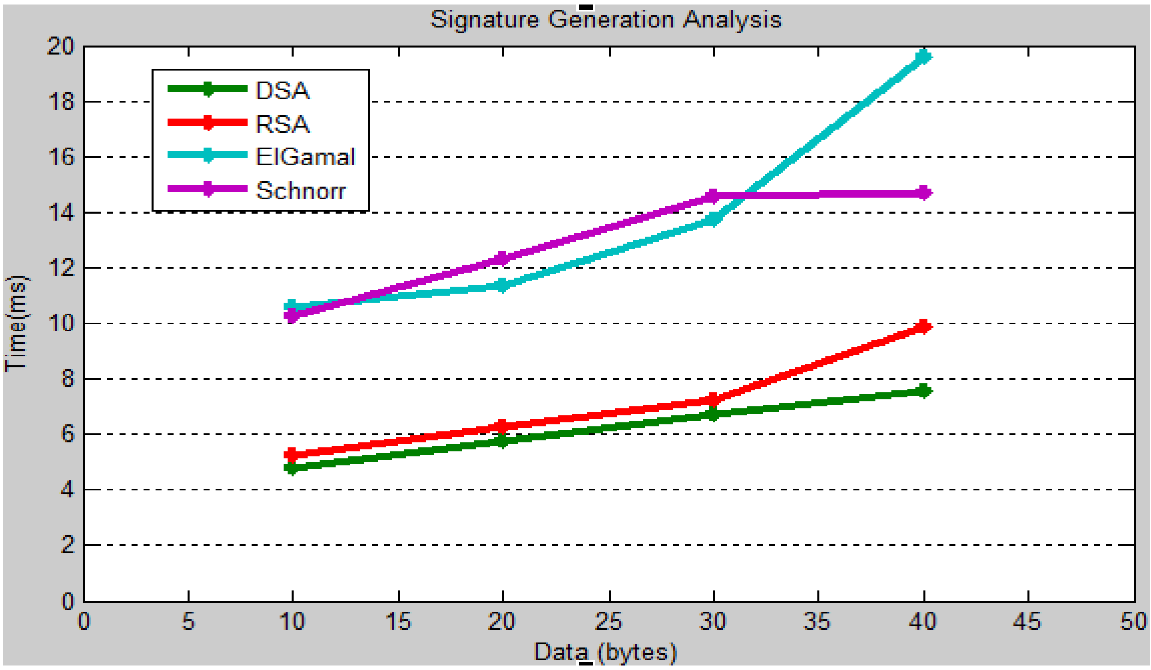

Fig. 2: A graph showing difference in Signature Generation on all algorithms

By comparing the digital signature scheme based on the time required to generate a digital signature, it can be deduced that the DSA implements at a much shorter time than the other algorithms making it the fastest of them all. It is therefore recommended that in situation where time is critical, as in a real time situation, in producing a digital signature, DSA should be considered. In the digital signature generation process, the private key is used to sign only the message digest. The signed message digest becomes the digital signature. From the graph pictured in figure 3, it is can also be deduced that Schnorr and ElGamal may progress at the same rate but the size of the data in question increases, Schnorr may prove to be a better choice to implement than ElGamal.

result of the plotted values is shown in the graph below

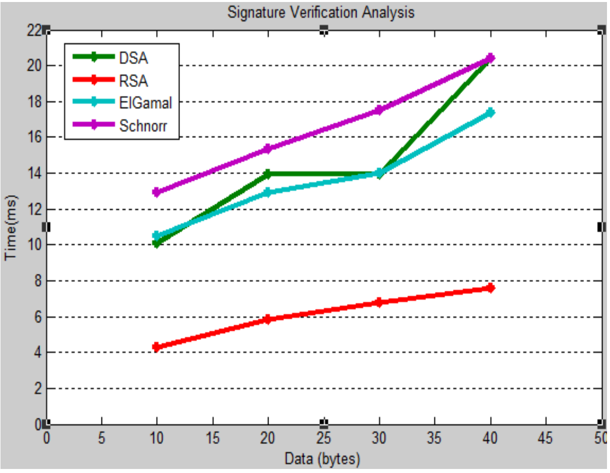

Fig. 3: A graph showing difference in Signature Verification time on all algorithms.

Table 2: Signature Verification Time based on data size

S/N	**Data size in Bytes**	**Digital Signature Algorithms Performance time in Milliseconds (ms)**			
		DSA	**RSA**	**ElGamal**	**Schnorr**
1	10	10.10	4.28	10.49	12.90
2	20	13.95	5.87	12.90	15.38
3	30	13.96	6.75	14.01	17.52
4	40	20.43	7.58	17.35	20.43

For Table 2, data measured in bytes was plotted on the x-axis while time measured in milliseconds can be seen on the y-axis. Random values were chosen and executed with other variables in stated in each of the algorithm procedures for verifying a signature on an encrypted message. The

The results are slightly different when the focus is Signature Verification using the same data size parameters as earlier stated in Figure 2, depicted in the table above. RSA proves to be an effective choice due to the lesser amount of time spent in carrying out verification of the signature on the message sent. This will be a key feature to note as it is recommended that signature verification be carried out online as offline implementation may be exposed to forgery.

The Analysis and Performance Evaluation of the Selected Digital Signature Schemes

From the analysis of the selected algorithms, it is seen that an increase in the number of bytes causes a corresponding increase in the running time as depicted in the graph. Though all the algorithms in this study are effective in securing data and information, it can be said that DSA is a more efficient algorithm in terms of performance time for signature generation while RSA has more efficiency in terms of signature verification.

CONCLUSION

It is therefore recommended that RSA be considered in an online/real-time situation where timing is of the essence. These findings are based on the evaluation of the selected algorithms. There is also a possibility of change in the values when analysis is carried out on a system with higher or lower specifications.

REFERENCES

Chu J., Xu Y., Li X., Lai Z. (2003). Research on Computing IP Core for the Digital Signature Algorithm. *Institute of Microelectronics Circuits and Systems, East China Normal University, Shanghai.* 2003,pp.1329 – 1331, Institute of Electrical and Electronic Engineers, ISBN: 0-7803-7889-X/03.

Etzel, B. (1995). New Strategy and Techniques to cope with Information Overload. *Proceedings of the IEEE Colloquim on Information Overload,* London, November 1995, pp.2/1–210, Institute of Electrical and Electronics Engineering.

Gilani J., Mir A. A. (2009). Using Digital Signature Standard Algorithm to incorporate non-invertibility in Private Digital Watermarking Techniques. *Proceedings of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (ACIS'09)*,2009,pp.399 – 404, Institute of Electrical and Electronic Engineers, ISBN: 978-1-7695-3642-2.

Helmy T., Amamiya S., Mine T., Amamiya M. (2003). A New Approach of the Collaborative User Interface Agents. *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)*, Halifax, Nova Scotia, UK, October 2003,pp.147 – 153, Institute of Electrical and Electronic Engineers, ISBN: 0-7695-1931-8.

Haddaji R., Ouni R. (2016). Comparison of Digital Signature Algorithm and Authentication Schemes for H.26 Compressed Video. *International Journal of Advanced Computer Science and Applications (IJACSA 2016),* Vol 7 (9) , pp.357 – 361, ResearchGate.

Xiao-fei L., Xuan-jing S., Hai-peng C. (2010). An Improved ElGamal Digital Signature Algorithm Based on Adding a Random Number. *Proceedings of the*

International Conference on Networks Secuirty, Wireless Communications and Trusted Computing,2010,pp.236 – 240, Institute of Electrical and Electronic Engineers, ISBN: 978-0-7695-4011-5.

Panjwani B., Deval C. M. (2015). Hardware-Software Co-design of Elliptic Curve Digital Signature Algorithm over Binary Fields. *Proceedings of the 2015 International Conference on Adavances in Computing, Communications and Informatics (ICACCI),* Kochi, India, August 2015, pp.1101 – 1106, Institute of Electrical and Electronic Engineers, ISBN: 978-1-4799-8792-4.

Rivest R., Shamir A., Adleman L. (1978). A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the Association for Computer Machinery, vol. 21* ,pp.120– 126, 1978.

Xiao Z., Wang Y., Jiang Z. (2015). Research and Implementation of Four-Prime RSA Digital Signature Algorithm. *Proceedings of the (ICIS'15)*, Las Vegas, USA, June/July 2015, Institute of Electrical and Electronic Engineers, ISBN: 978-1-4799-8679-8.