Enhanced Secured Wireless Message Communication using Digital Signature Algorithm (DSA)

Ade, Sandra Ngi[1], Eru, Nathaniel Akwuma[2], Nyamtswam, Ngunengen[3] and Ayangeaor, Raphael Terkende[4]

[1,2,3,4]Postgraduate School, College of Science, Department of Mathematics/Statistics/Computer Science, Federal University of Agriculture, Makurdi, Benue State, Nigeria.

Abstract

This paper focuses on how to protect communications that occur in a transaction so as to guide against fraudsters. Digital Signature Algorithm (DSA) is adopted to be used with the Secure Hash Algorithm-2 (SHA-2). The security of the DSA is based on the difficult computable nature of discrete logarithm over finite fields; as such, the current key size of the DSA is small and thus makes it easy for an attacker to launch a brute force attack. In order to counter this problem, the size of the prime numbers that form the basis of the key generation process used in the DSA has being increased. In Digital Signatures, the message sent to the recipient is digitally signed, but not encrypted hence, the message's confidentiality is not guaranteed. This work therefore incorporates encryption to a digitally signed message to ensure the message's absolute privacy. This eliminates the possibility of eavesdropping which also improves the security of the DSA. An object oriented approach is used to model the new system. The implementation of the system is done using Java Enterprise Java Bean 3 at the front end and MySQL database server is used at the back end for persistent storage of messages, private and public keys running on JBOSS server.

Keywords - *Cryptography, DSA (Data/Digital Signature Algorithm, SHA (Secure Hash Algorithm), MD (Message Digest), Cryptosystem, Encryption, Secret/Private Key, Public Key, RSA.*

I. INTRODUCTION

The progressive widespread of wireless networks has led many offices and other public places to make use of them as a means of communication today. They are suitable due to their portability and high-speed information exchange in homes, offices and enterprises. The advancement in modern wireless technology has led users to switch to a wireless network compared to a Local Area Network (LAN) connected using Ethernet cables.

In wireless networks, large amount of data is generated, which are often sensitive and vulnerable to interceptions than wired networks. These security threats have increased as the service is now more popular. Initially, there were a few dangers when wireless technology was first introduced because hackers had not yet understood the then new technology when wireless networks were not commonly found in workplaces. However, there are many security risks associated with the current wireless protocols and encryption methods; and as a result of the carelessness and ignorance that exist at the user and corporate IT levels. To combat this problem, wireless network users may choose to utilize various encryption methodologies.

Reference [1] defined encryption as an enabling technology that provides companies, their business partners, customers and end users with the ability to get the information and service they need much faster and more securely. When information is encrypted, it is scrambled into a code so that others cannot comprehend it. Due to the high possibility of information compromise accompanying wireless networks, various encryption methods have been developed, many of which are known to have weaknesses and are susceptible to attacks thereby compromising confidentiality [2].

In order to establish secure communications around a wireless network, it is important that communication between nodes (users) and base stations to other nodes should be handled carefully by means of an efficient Cryptographic Algorithm. Cryptographic algorithms can be classified into two types; these are symmetric key and asymmetric key cryptographic algorithms. In Symmetric Key Algorithm, a single key is used to encrypt and decrypt the data while, Asymmetric Key Algorithm uses two types of keys; Public Key for the encryption and Private Key for the decryption [3]. Symmetric (Secret) key cryptography includes Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES, International Data Encryption Algorithm (IDEA), Blowfish Algorithms, etc. and public key cryptography includes RSA, Digital Signature Algorithm (DSA), Deffie-Hellman Algorithm, Elliptic Curve Cryptography (ECC), etc [4] [5].

Two important properties of cryptosystems are its speed and security. Speed denotes the time taken by the algorithm to convert a given plain text to cipher text. Key plays a projecting role in encryption and decryption of the algorithm. Its size determines the strength of encryption algorithms. The increase in key size reduces the speed of the algorithm but in turn increases the security.

In this study, DSA is adopted. Digital signature is a public key cryptosystem which is used to provide authentication, non-repudiation, and data integrity services to information [6].

A digital signature scheme typically consists of three algorithms: a key generation algorithm that selects a private key uniformly at random from a set of possible private keys – the algorithm outputs the private key and a corresponding public key; a signing algorithm that, given a message and a private key, produces a signature; a signature verifying algorithm that, given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

Although the DSA provides authentication, non-repudiation, data integrity services to information, it however does not provide confidentiality of the exchanged information. As the data exchanged is not encrypted. Therefore, if data is intercepted, the intruder can easily have access to the information.

Another problem of the DSA is that the signature generation phase of the DSA uses Secured Hash Algorithm-1 (SHA-1) to generate hash values for an input string. The SHA-1 has been proven to have a likelihood of producing collisions hence it is possible for the SHA-1 to produce the same hash values for two different messages. This weakness can thus be exploited by an attacker when he compares the hash values together. Therefore, it is no longer safe for the DSA to continue with its use.

Furthermore, the primes which form the basis for the generation of the cryptographic keys of the DSA need to be increased to larger size. This is because the current key sizes no longer guarantee the security of the DSA.

This work is therefore aimed at improving the security aspect of the DSA by incorporating encryption mechanism to the message sent over the network so as to provide confidentiality and also to enhance the key strength of the cryptographic keys – the size of the primes – ***p*** and ***q*** used for the key generation to be of 256 and 2048 bits respectively and to remove the possibility of brute force attack by the use of SHA-2 (rather than SHA-1) to generate unique hash values for each message being sent over the network.

$$\boxed{y = g^{x} \bmod p.}$$

II. LITERATURE REVIEW

A. *Digital Signature Algorithm (DSA)*

The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard (FIPS) 186, known as the Digital Signature Standard (DSS). The DSS makes use of the Secure Hash Algorithm (SHA) and presents a new digital signature technique, the Digital Signature Algorithm [7]. The SHA takes an arbitrary length string and returns a fixed-length hash value output which is called the Message Digest (MD). Signing the MD rather than the message often improves the efficiency of the process because the MD is usually much smaller in size than the original message [8]. Strong cryptographic hash functions make it easy to compute the hash value from an input string but very difficult to find a message that results in a particular hash value, modify a message without changing its hash value, or find two messages that hash to the same value. DSAs rely on cryptographic hash functions.

The DSS was originally proposed in 1991 and revised in 1993 in response to public feedback concerning the security of the scheme. There was a further minor revision in 1996. In 2000, an expanded version of the standard was issued as FIPS 186-2, subsequently updated to FIPS 186-3 in 2009. The latest version, FIPS 186-4 issued in July, 2013 specifies the use of larger key sizes for a stronger DSA [9].

B. *Phases of DSA*

1. *Key Generation*

The key generation has two phases. The first phase is the choice of algorithms parameters (***p***, ***q***, and ***g***) which may be shared between different users of the system. This involves choosing an approved cryptographic function H, such as SHA-1. An *L*-bit prime modulus ***p*** and *N*-bit prime ***p*** which determine the measure of the cryptographic strength of the key. The original DSS constrained *L* to be between 512 and 1,024 (inclusive). NIST 800-57 recommends lengths of 2,048 (or 3,072) for keys with security lifetimes extending beyond 2010 (or 2030) [9]. To end the first phase, *g* is chosen – an integer whose multiplicative order modulo ***p*** is ***q***. This may be done by setting $\boldsymbol{g = h^{(p-1)/q} \bmod p}$ for some arbitrary *h* (1 < *h* < *p*−1), and trying again with a different ***h*** if the result comes out as 1.

The second phase computes private and public keys for a single user given a set of parameters. The private key ***x*** must be a number from 1 to (*q-1*) and is chosen randomly or pseudo-randomly. The public key ***y*** is calculated from the private key ***x*** as;

The calculation of ***y*** given ***x*** is relatively straightforward. However, given the public key ***y***, it is believed to be computationally infeasible to determine the private key ***x***, which is the discrete logarithm of ***y*** to the base ***g***, mod ***p***.

2. *Signature Generation*

The signing algorithm is the second phase of the digital signature scheme. During this process, the plain text (message) is digested using the SHA-1 to produce a 160-bit MD (hash code). A user then calculates two quantities, ***r*** and ***s***, which forms the signature and are functions of the public key components (*p*, *q*, and *g*), the user's private key ***x***, the hash code of the message H(*m*), and an additional integer ***k*** that should be generated randomly or pseudo-randomly and be unique for each signing.

Signature = (r, s)

3. *Signature Verification*

The signature verification algorithm, which is the third and last phase of the digital signature scheme, is executed at the recipient's end. The receiver collects the message and the signature transmitted by the sender; then obtains the sender's public key to verify the signature of the received message. This is done using the following formula:

$$\mathbf{w = (}s'\mathbf{)^{-1} \bmod} \ q$$

$$\mathbf{u1 = [H(m') \, w] \bmod} \ q$$

$$\mathbf{u2 = (}r'\mathbf{) \, w \bmod} \ q$$

$$\mathbf{v = [(}g^{u1} \, y^{u2}\mathbf{) \bmod} \ p\mathbf{] \bmod} \ q$$

TEST: v=r

Where;

w is called the modular multiplicative inverse of *s* modulo *q.*

m is the message to be signed.

H(m): is the hash of ***m*** using SHA-1

m´, **r´**, **s´** is the received versions of **m**, **r**, **s.**

v is a function of the public key components, the sender's public key, and the hash code of the incoming message.

The receiver generates a quantity ***v***, if this quantity matches the ***r*** component of the signature, then the signature is validated.

The test at the end is on the value ***r***, which does not depend on the message at all. Instead, ***r*** is a function of ***k*** and the three global public-key components. The multiplicative inverse of ***k*** (mod ***q***) is passed to a function that also has, as inputs, the message hash code and the user's private key. The structure of this function operates in such a way that the receiver can recover ***r*** using the incoming message and signature, the public key of the user, and the global public key.

C. *Secure Hash Algorithm 1 (SHA-1)*

A hashing algorithm is a function that takes in an arbitrary length block of data, and returns a fixed-size string, which is called the hash value or MD [10]. The hash functions which have been popularly used by the DSA and most other applications have been the MD4, MD5, SHA-0, and SHA-1.

The hash function SHA-1 was issued by NIST in 1995 as a FIPS. Since its publication, SHA-1 has been adopted by many governments and industries as security standards; in particular, standards on digital signatures for which a collision-resistant hash function is required [11]. In addition to its usage in digital signatures, SHA-1 has also been deployed as an important component in various cryptographic schemes and protocols, such as user authentication, key agreement and pseudorandom number generation having replaced MD5 as the secure hash function of choice when a number of security flaws were discovered in MD5. Subsequently, SHA-1 has been extensively implemented in almost all commercial security systems and products.

Digital signatures make use of asymmetric cryptographic operations to prove that a message was signed by someone in possession of the corresponding private key. However, asymmetric cryptographic operations are computationally expensive in terms of both key size and as well as the length of the input. Because of this, digital signatures sign a hash of the message instead of the message itself. As long as the hash function is a "secure" hash function, this is sufficient. It will thus be computationally impracticable for the attacker to create another message that has the same hash value.

Breaking this property requires finding two messages that share the same hash. This is called a *collision attack*. Once an attacker has two messages that share the same hash, he can create a forged message and get it signed and it will validate correctly. This attack was used to create forged MD5 certificates, most famously back in 2012 by the Flame malware to create fraudulent Microsoft certificates that could be used to fool Windows Updates [12]. Because of the birthday paradox, this happens when the number of messages is approximately the square root of the total number of possible hashes [13]. Therefore, when considering collision resistance, a hash function has an equivalent strength of at most half the number of bits in the hash, and possibly fewer. Since SHA-1 produces a 160-bit hash, the strength is at most 80 bits.

In 2013, building on these advances and a novel rigorous framework for analysing SHA-1, the state-of-the-art collision attack on full SHA-1 was presented by Stevens with an estimated 61 bits of effective strength. He further pointed out that cryptographic advances will continue to reduce the strength even further [14].

Several Digital signature schemes, such as RSA, Elliptic Curve Cryptography (ECC), ElGamal, etc. have been proposed to provide authentication over public networks using asymmetric cryptography. According to [15], the DSA is an efficient variant of ElGamal and the ElGamal has several drawbacks which the DSA repairs. Some of these are;

- Given today's security standards, an ElGamal Signature is of bit length at least 2048. A DSA signature is of bit length 320.
- Signature verification in the ElGamal scheme requires three modular exponentiations with exponents of bit length at least 1024. Signature

verification in the DSA requires only two modular exponentiations with exponents of bit length 160.

- Some of the attacks that can threaten ElGamal scheme are not applicable to DSA.

Although the DSA is an efficient digital signature, given current algorithmic knowledge, the DSA appears to be secured for now. Its security depends on the difficulty of finding discrete logarithms. The size of the parameters (*p*, *q*, *and g*) also plays an essential role in the security of the DSA. The parameters represent;

- p – a large prime number (at least 1024 bits).
- q – a sufficiently large prime number (at least 160 bits) that is also a divisor of *p-1*.
- g – a number whose multiplicative order modulo *p* is *q* given as;

QUOTE
Where;
QUOTE is an integer between 1 and p-1

Reference [16] suggested that, the DSA should be used with larger primes for longer security periods in an environment where it can be easily replaced by another signature scheme, if necessary.

NIST provides Guidelines for Public-Key Sizes, which shows that the size of RSA and DSA keys grows at a much faster rate than those based on ECC when faced with increasing security requirements [17]. Key length is directly proportional to security strength. It is measured in bits and each bit of key exponentially increases the difficulty of a brute-force attack. It is important to note that in addition to adding more security, each bit slows down the cryptosystem as well.

Furthermore, it is observed that the key sizes of DSA are identical to RSA and that, key generation of digital signature using DSA is faster. Key verification is slightly slower. DSA is also compatible with most servers, and because it is already a federal standard, using Secure Sockets Layer (SSL) certificate makes it easier for businesses to meet the requirements of government contracts.

III.THE PROPOSED SYSTEM

A. *Secure Hash Algorithm-2 (SHA-2)*

In the aftermath of the SHA-1 attacks, the advice NIST produced was to move to SHA-2 [18]. As a result, many standards and products have started to move towards larger hash sizes. This new family of hashing algorithms known as SHA-2 which comprises of SHA-224, SHA-256, SHA-384 and SHA-512, uses larger hash values, making them more resistant to imminent attacks. SHA-256 produces a 256-bit hash value. That is, every message hashes down to a 256-bit number. Given that there are an infinite number of messages that hash to each possible value, there are an infinite number of possible collisions. But because the number of possible hashes is so large, the odds of finding one by chance has not yet been discovered by employing cryptanalysis. Using birthday problem, a hash function that produces n-bits is broken in $2^{n/2}$ time which is much faster than using the brute-force attack. The SHA-1 hash functions have been shown to be vulnerable to the birthday attacks.

In this research work, the SHA-256 of the SHA-2 family has been used to produce a hash value of an input message. The following are the properties of SHA-2 which gives it its security and therefore makes it useful in the enhancement of the DSA.

1. Pre-Image Resistance

It is computationally hard to reverse a SHA-2 function. In other words, when SHA-2 produces a hash value ***z***, it is difficult to find any other input value ***m*** that hashes to ***z***. This property protects against an attacker who only has a hash value and is trying to find the input.

2. Second Pre-Image Resistance

Given an input and its hash, it is hard to find a different input with the same hash. This implies that, for an input ***m*** (message 1), SHA-2 produces a hash value **H(*m*)**, which is difficult to find any other input ***y*** (message 2) such that h(*y*) = H(*m*). This property of SHA-2 protects against an attacker who has an input value and its hash, and wants to substitute a different value as legitimate value in place of original input value.

3. Collision Resistance

This property means that, it is hard to find two different inputs of any length that result in the same hash. This property is also referred to as *collision-free hash function* [19], which implies that, it is hard to find any two different inputs ***m*** and ***n*** such that H(*m*) = H(*n*) using the SHA-2. This property makes it very difficult for an attacker to find two input values with the same hash.

B. *Algorithm for Key Generation*

In the new system, the sizes of the keys are increased to make the system more secured. This is achieved by increasing the size of the prime numbers used in the generation of the private and public keys; ***p*** and ***q*** represents the prime numbers. The parameters (***p***, ***q***, ***and g***) are public and can be common to a group of users; ***q*** is a 256 bit prime number which is chosen, followed by the selection of ***p*** – a prime number with length of 2048 bits, and ***g*** is an integer whose multiplicative order modulo ***p*** is ***q*** and is calculated by the formula;

```
STEP 1: Choose a prime number q of 256 bits
STEP 2: Choose a prime number p of 2048 bits
such that (p-1 mod q = 0)
STEP 3: Compute g
            $g = h^{(p-1)/q} \bmod p$
STEP 4: Choose a private key x between 1 to
q-1
STEP 5: Calculate the public key y
            $y = g^{x} \bmod p.$
```

$$g = h^{(p-1)/q} \bmod p.$$

C. Flowchart of a DSA

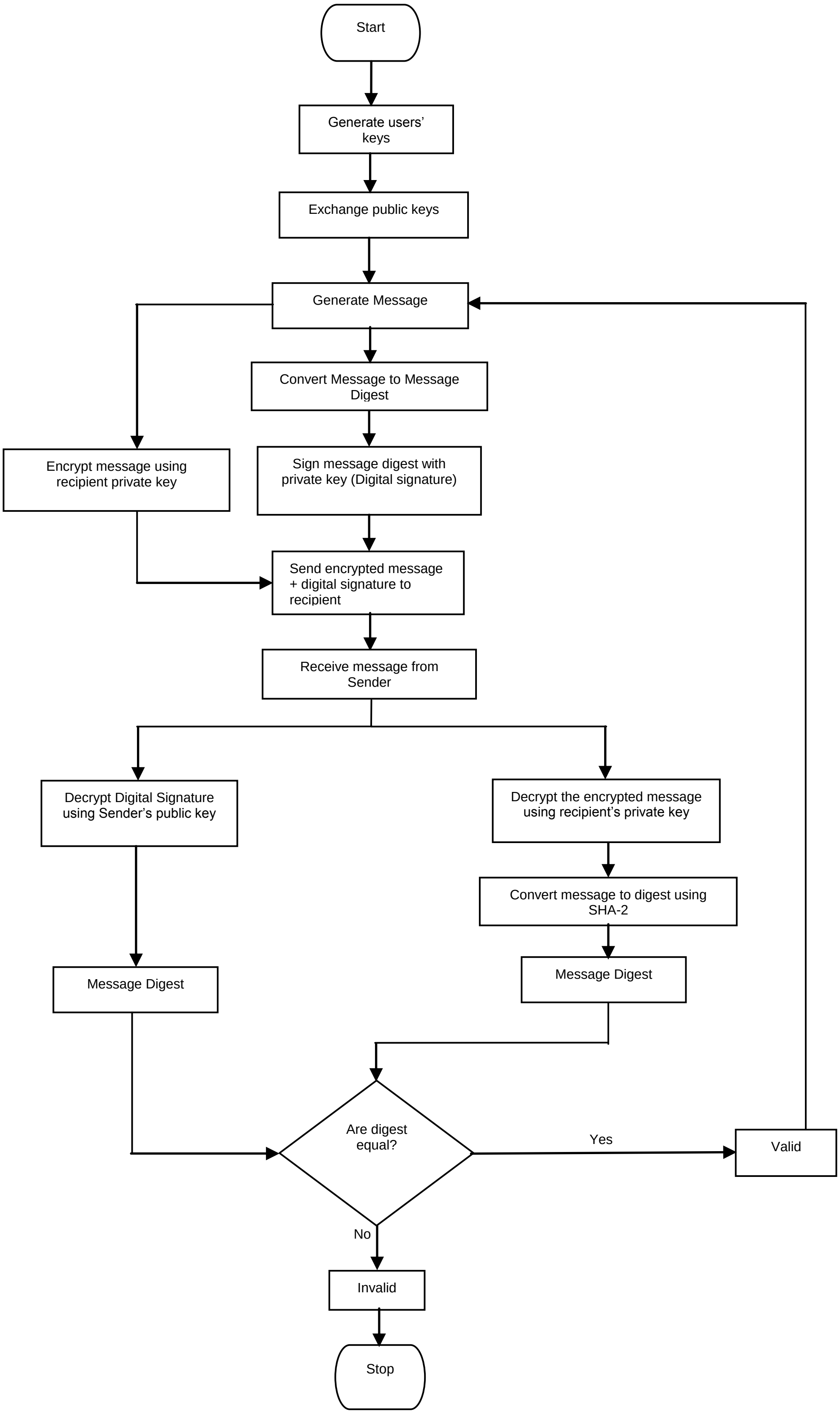

Start
Generate users' keys
Exchange public keys
Generate Message
Convert Message to Message Digest
Encrypt message using recipient private key
Sign message digest with private key (Digital signature)
Send encrypted message + digital signature to recipient
Receive message from Sender
Decrypt Digital Signature using Sender's public key
Decrypt the encrypted message using recipient's private key
Convert message to digest using SHA-2
Message Digest
Message Digest
Are digest equal?
Yes
Valid
No
Invalid
Stop

Fig 1: Flowchart of a Digital Signature Algorithm

The algorithm below shows how this is achieved.

D. How the System Operates

In Fig 2, the sender digests the message with SHA-2 which produces the MD, h(***m***). He then signs the MD with his private key which produces the Digital Signature (S). The Digital Signature is then appended to the encrypted message and sent to the recipient. At the receiver's end, the digital signature is decrypted with the help of the sender's public key corresponding to the private key that was used during the signing process, thus producing an MD.

The recipient also decrypts the encrypted message received with the help of his private key corresponding to the public key used in encrypting the message. The decrypted message (plain text) is passed through the same hashing algorithm that was used during the signing process which now gives the MD. The verification of digital signature is done by comparing the two MDs or hash values. If the two values are the same, the verification is successful and proves that the message has been signed with the private key that corresponds to the public key used in the verification process and was not altered during transit; hence, valid. If the two values are different, this means that the message may have been altered by a third party during transit; thus making the digital signature invalid and the verification is unsuccessful.

E. Architecture for Signature Generation and Verification

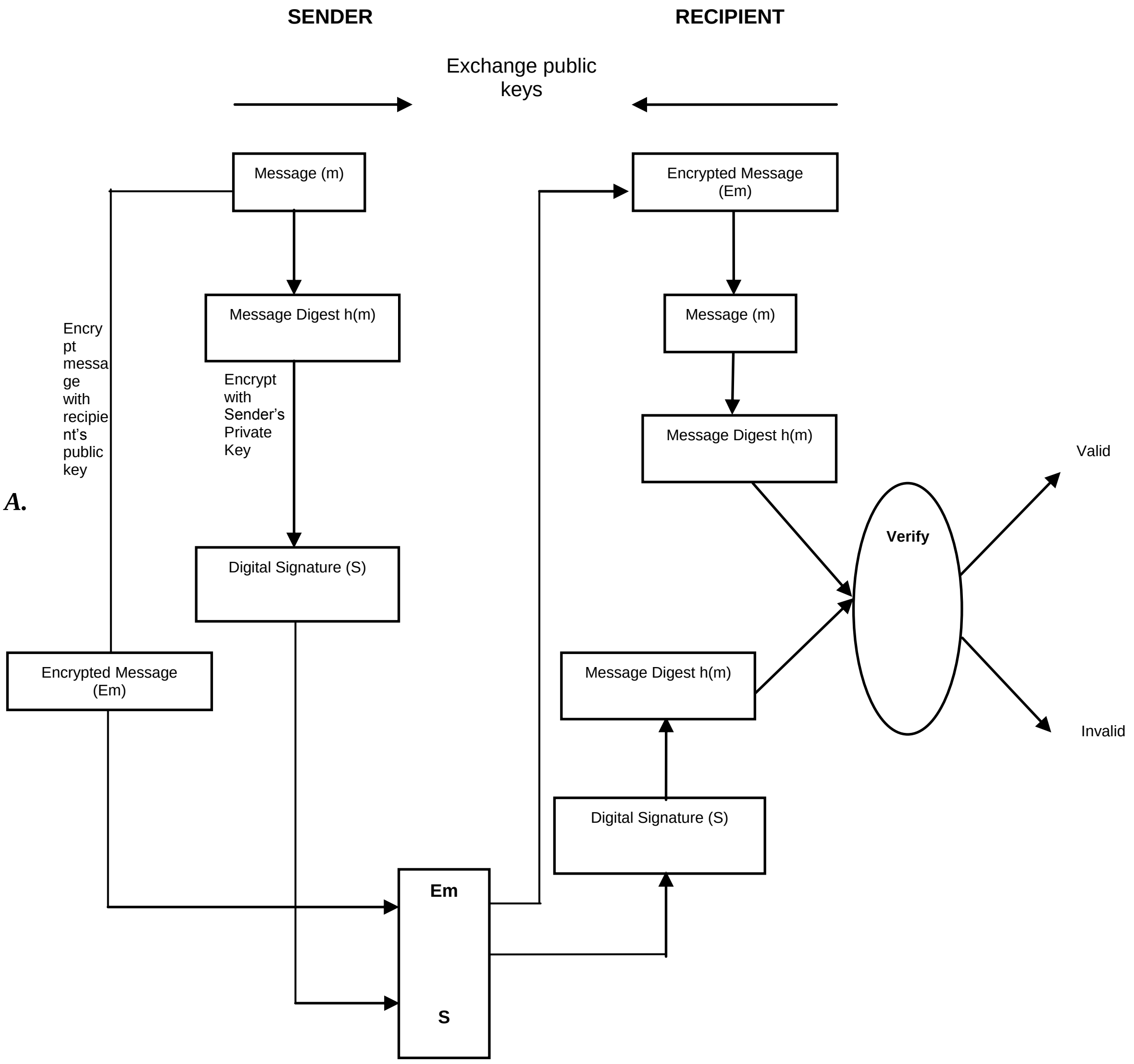


Fig 2: Architecture for the process of signature generation and verification of DSA

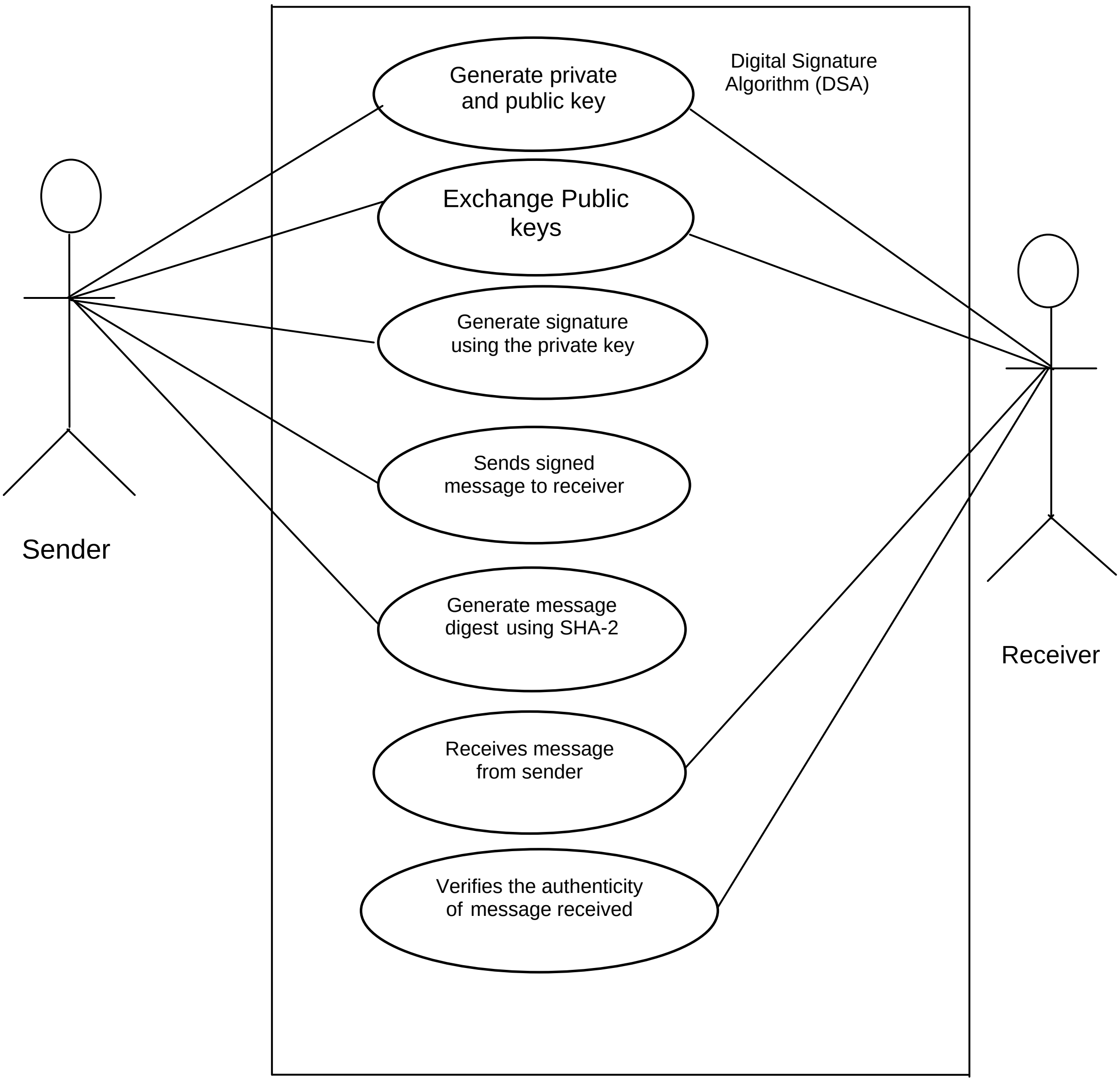

Digital Signature Algorithm (DSA)
Generate private and public key
Exchange Public keys
Generate signature using the private key
Sends signed message to receiver
Generate message digest using SHA-2
Receives message from sender
Verifies the authenticity of message received
Sender
Receiver

Fig 3: A Use-Case diagram of DSA describing the functions of the objects in the system

IV.RESULTS AND DISCUSSION

A. Implementation

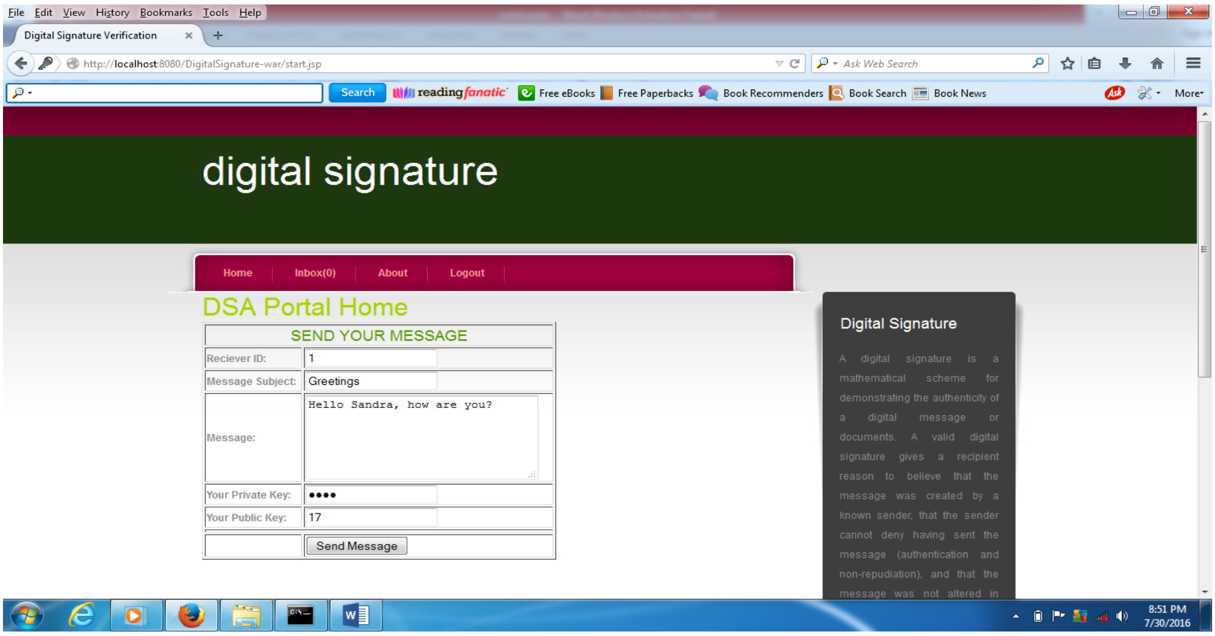
The implementation of this system is done using Java Enterprise Java Bean 3. In addition to that, other system requirements include Java Development Kit (JDK version 7), Java Virtual Machine (JVM), and JBOSS Application Server which is the localhost server on which the application is deployed.

How the system works: The application runs on a web browser since it is an enterprise solution. The system uses SHA-2 of the DSA to secure messages sent over the network.

The Digital signature is generated by taking two different inputs- the message and the signer's private key. When a user sends a message, the message will be digested using the SHA-2. The signature will be unique for the different sets of inputs.

B. Results of the Implementation and Discussion

In Fig 4, the sender generates the message to be sent to the recipient (whose ID is 1). He further supplies his private key with the corresponding public key; the private key which will further be used to generate the digital signature and the public key which is sent to the recipient. The public key will be used at the recipient's end on the receipt of the message to decrypt the signature.



DSA Portal Home

SEND YOUR MESSAGE

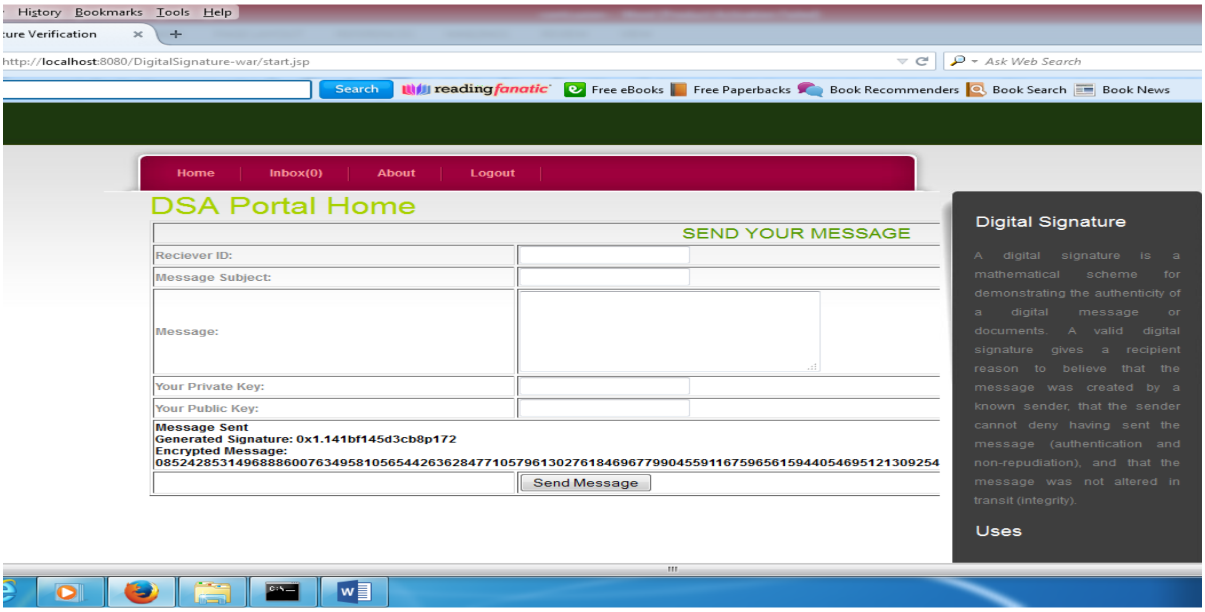
Reciever ID:	1
Message Subject:	Greetings
Message:	Hello Sandra, how are you?
Your Private Key:	••••
Your Public Key:	17
	Send Message

Digital Signature

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, that the sender cannot deny having sent the message (authentication and non-repudiation), and that the message was not altered in

Fig 4: A User Sending Messaging to another User

Fig 5 shows how the signature is generated and appended to the message which in turn is encrypted and ready to be sent to the recipient. This occurs as a result of converting the message generated by the sender in Fig 4 to a hash value and encrypting the hash value with the private key of the sender to produce the signature. The plain message is also encrypted with the public key of the recipient before being appended to the signature.



DSA Portal Home

SEND YOUR MESSAGE

Reciever ID:	
Message Subject:	
Message:	
Your Private Key:	
Your Public Key:	

```
Message Sent
Generated Signature: 0x1.141bf145d3cb8p172
Encrypted Message:
08524285314968886007634958105654426362847710579613027618469677990455911675965615944054695121309254
```

Send Message

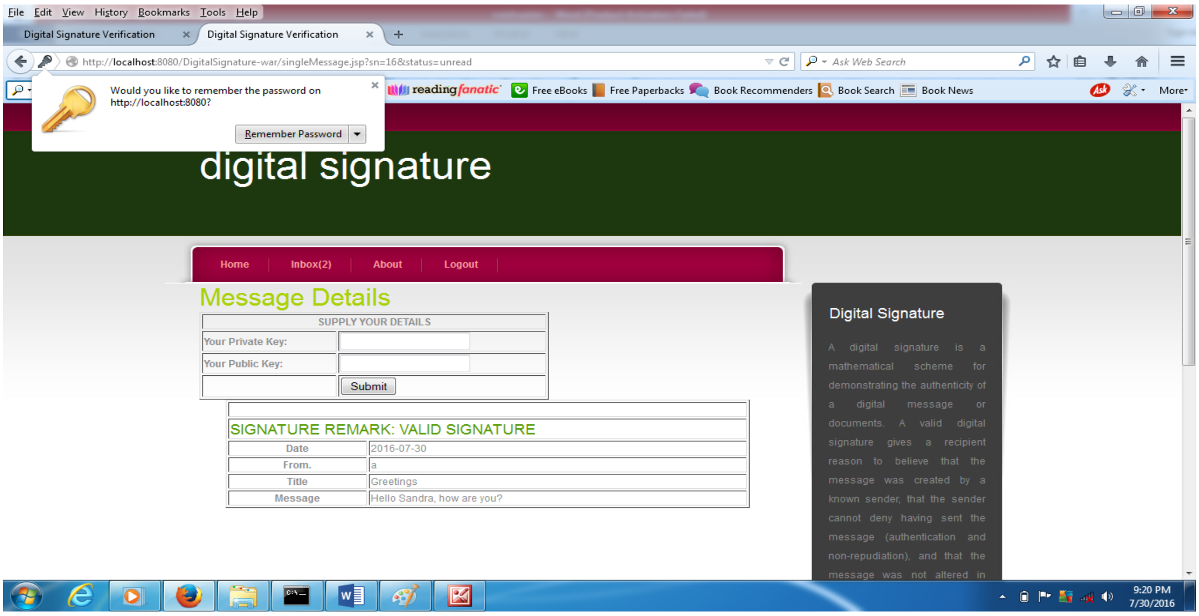
Digital Signature

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, that the sender cannot deny having sent the message (authentication and non-repudiation), and that the message was not altered in transit (integrity).

Uses

Fig 5: Signature Generation and Message Encryption

In Fig 6, the recipient, on receipt of the message, performs verification to prove the authenticity of the message. The recipient inputs his private key which will be used to decrypt the message and then the public key of the signer which will be used in the verification process in the program code. At the end of the verification process, the signature remark shows valid signature which means, the message was not altered by a third party in the course of transit, and the public key which corresponds to the private key that was used in signing the message is used for the verification process.


File Edit View History Bookmarks Tools Help
Digital Signature Verification
Digital Signature Verification
http://localhost:8080/DigitalSignature-war/singleMessage.jsp?sn=16&status=unread
Ask Web Search
reading fanatic
Free eBooks
Free Paperbacks
Book Recommenders
Book Search
Book News
More
Remember Password
Home
Inbox(2)
About
Logout
Submit
9:20 PM
7/30/2016

Would you like to remember the password on http://localhost:8080?

digital signature

Message Details

SUPPLY YOUR DETAILS

Your Private Key:

Your Public Key:

SIGNATURE REMARK: VALID SIGNATURE

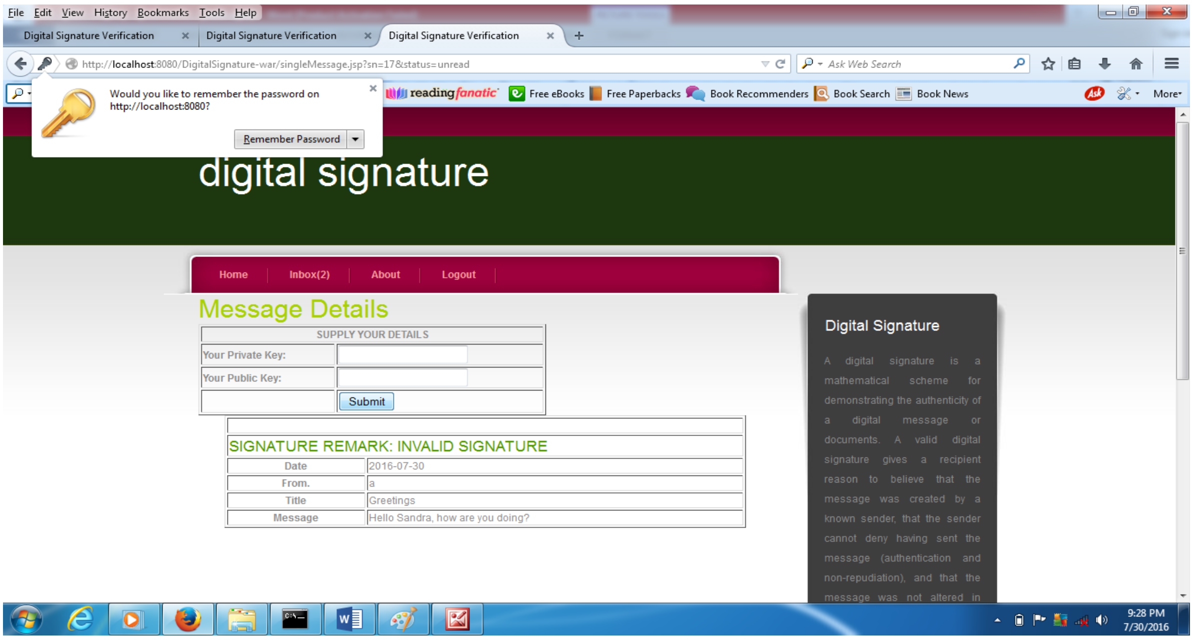
Date	2016-07-30
From.	a
Title	Greetings
Message	Hello Sandra, how are you?

Digital Signature

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, that the sender cannot deny having sent the message (authentication and non-repudiation), and that the message was not altered in

Fig 6: Verification of Signature (Valid)

Fig 7 shows an invalid signature for the message sent. This proves that the message was not signed by the private key which corresponds to the public key used in the verification process. This happened when an attack on the message occurred. An attacker who had access to the public key decrypted the signature producing its hash. The hash value was encrypted again by the attacker using another private key (unknown to the recipient); this is because the attacker had neither access to the sender nor recipient's private key. He then sends it to the intended recipient which on verification of the signature proves invalid.


File Edit View History Bookmarks Tools Help
Digital Signature Verification
Digital Signature Verification
Digital Signature Verification
http://localhost:8080/DigitalSignature-war/singleMessage.jsp?sn=17&status=unread
Ask Web Search
reading fanatic
Free eBooks
Free Paperbacks
Book Recommenders
Book Search
Book News
More
Remember Password
Home
Inbox(2)
About
Logout
Submit
9:28 PM
7/30/2016

Would you like to remember the password on http://localhost:8080?

digital signature

Message Details

SUPPLY YOUR DETAILS

Your Private Key:

Your Public Key:

SIGNATURE REMARK: INVALID SIGNATURE

Date	2016-07-30
From.	a
Title	Greetings
Message	Hello Sandra, how are you doing?

Digital Signature

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, that the sender cannot deny having sent the message (authentication and non-repudiation), and that the message was not altered in

Fig 7: Signature Verification (Invalid)

V. CONCLUSION

In this paper we have been able to show how it can be ensured that a message sent across a network is secured using DSA. The SHA-2 has been used to enhance the security of the DSA by removing the collisions of hash values. The system is guaranteed to be collision resistant.

In addition to the security advantage, this system has been proved to maintain the confidentiality of information, since any message in transit is encrypted before appending to the signature. Also, the integrity of the information is securely assured since the length of the message is preserved during any of the operations of encryption and decryption. The system has therefore eliminated the fear of losing vital information to an eavesdropper or an enemy at any point in transmission time.

REFERENCES

[1] McNulty, F. (1999). Encryption’s Importance to Economic and Infrastructure Security. Retrieved from scholarship.law.duke.edu/cgi/viewcontent.cgi?article=1234&context=djcil.pdf.

[2] Bhatia, P., & Sumbaly, R. (2013). Framework for Wireless Network Security using Quantum Cryptography. Retrieved from https://arxiv.org/pdf/1412.2495.pdf.

[3] Yakubu, B. M. (2015). Advanced Secure Method for Data Transmission in Manet using RSA Algorithm. International Journal of Advanced Technology in Engineering and Science, 10.

[4] Kessler, G. C. (1999, May 13). An Overview of Cryptography. McGraw-Hill.

[5] Gupta, A., Patel, A., Kumar, L., & Tomar, D. (2017). Security System for DNS using Cryptography. International Journal of Engineering Trends and Technology (IJETT), 46 (9), 462–465.

[6] Krishna, D. S. (2015). Providing Security to Confidential Information Using Digital signature. International Journal of Advance Research in 1Computer Science and Management Studies, 5.

[7] Stallings, W. (2011). Cryptograpy and Network Security Principles and Practices (Fifth Edition). New York: Prentice Hall.

[8] Torres, A. (2013, 8 14). Retrieved from pythoncentral: http://pythoncentral.io/hashing-strings-with-python/

[9] Gallagher, P. D. (2013). Federal Information Processing Standards Publication- Digital Signature Standard (DSS). Gaithersburg, MD 20899-8900.

[10] Raheja, S., Verma, S., & Raheja, N. (2014). Review and Analysis of Hashing Techniques.

[11] International Journal of Advanced Research in Computer Science and Software Engineering, 296-298.

[12] Chang, S.J., Perlner, R., Burr, W. E., Turan, M. S., Kelsey, J. M., & Paul, S. (2012). Third-Round Report on SHA-3 Cryptographic Hash Algorithm Competition. United States: National Institute of Standard and Technology.

[13] Keizer, G. (2012, 06 05). Six Steps to a better Security Strategy. Retrieved from Computer World: www.computerworld.com/article/2503916/malware-vulnerabilities/researchers-reveal-how-flame-fakes-windows-update.html

[14] Simion, E. (2012, 12). The Birthday paradox. Operational Research and Optimization .

[15] Stevens, M. (2013). New Attacks on SHA-1 based on Optimal Joint Local- Collision Analysis. Lecture Notes in Computer Science, vol. 7881, Springer, pp. 245-261

[16] McCurley, S., & Gordon. (1999). Elliptic Elliptic Curves and Their Applications to Cryptography: An Introduction. https://books.google.com.ng/books?isbn=0792385896: Springer shop Amazon.com

[17] Buchman, J. (2001, 12 15). The Digital Signature Algorithm (DSA). Retrieved from https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1003_DSA.pdf

[18] Greg, C. (2011). Enhancing Website security with Algorithm Agility. Symantec

[19] Co-operation.

[20] Chang, S.j., Perlner, R., Burr, W. E., Turan, M. S., Kelsey, J. M., & Paul, S. (2012). Third-Round Report on SHA-3 Cryptographic Hash Algorithm Competition. United States: National Institute of Standard and Technology.

[21] Schneier, B. (2012). Applied Cryptography (2nd Edition). John Wiley & Sons.