ISSN 2229-5518

Make a Secure Connection Using Elliptic Curve Digital Signature

H. Modares, M. T. Shahgoli, H. Keshavarz, A. Moravejosharieh, R. Salleh

AbstractIt is generally accepted that data encryption is the key role in current and future technologies. Many Public key cryptography schemes were presented, divided into different classes depending on a specific mathematical problem. Cryptography plays an important task in accomplishing information security. It is used for encrypting or signing data at the source before transmission, and then decrypting or validating the signature of the received message at the destination. Since the introduction of public-key cryptography by Diffie and Hellman in 1976, the potential for using the discrete logarithm problem in public-key cryptosystems has been recognized. There are several public key cryptography, such as RSA and El-Gamal and Elliptic curve cryptography. Elliptic Curve Cryptography (ECC) is considered more suitable than other public key cryptography algorithms because of its small key size. ECC is chosen in this work because of its advantages over other public key cryptography. Generally, to produce private keys and elliptic curve cryptography domain parameters, a random generator is used.

Index Terms—RSA, Elliptic Curve Cryptography (ECC), public key cryptography (PKC), Elliptic Curve Digital Signature (ECDSA), Hashing

1 INTRODUCTION

Cryptography is the science of hiding information which can be revealed only by legitimate users. It is used to ensure the secrecy of the transmitted data over an unsecure channel and prevent eavesdropping and data tampering. Another field called cryptanalysis that is concerned of attacking and decrypting this ciphers.

Many cryptography schemes were proposed and used for securing data, some uses the shared key cryptography and some uses the public key cryptography (PKC). Shared key cryptography is a system that is uses only one key by both sender and receiver for purpose of encrypting and decrypting the message. On the other hand, public key cryptography uses two keys, private-key and public-key. To encrypt a message in Public key scheme, public-key will be used and to decrypt it back a private-key is used.

As compared to the shared key cryptography, public key cryptography are slow. However, public-key cryptography can be used with shared key cryptography to get the best of both. Public key cryptography have many advantages over the shared key, it increases the security and convenience where there is no need to distribute the private key to anyone.

Elliptic curves are algebraic curves that have been studied by many mathematicians for long time. In 1985 Neal Koblitz and Victor Miller independently proposed public key cryptosystems using elliptic curve. Since then, many researchers have spent years studying the strength of ECC and improving techniques for its implementation. Elliptic curve cryptosystem (ECC) provides a smaller and faster public key cryptosystem.

ECC has been commercially accepted, and has also been adopted by many standardizing bodies such as ANSI, IEEE, ISO and NIST. ANSI in their standard provided the needed algorithms to generate an elliptic curve and generating Elliptic Curve Digital Signature (ECDSA) signatures. It provides step by step examples to generate and verify ECDSA for differing key sizes.

Elliptic curves defined over finite fields which provide a group structure that is used to implement the cryptographic schemes. Scalar point multiplication is the major building block of all elliptic curve cryptosystems, an operation of the form $k \cdot P$ where k is a positive integer and P is a point on the elliptic curve. Calculating $k \cdot P$ gives the result of adding the point P to itself for exact k-1 times, which results in another point Q on the elliptic curve. The inverse operation, i.e., to recover k when the points P and $Q = k \cdot P$ are given is known as the Elliptic Curve Discrete Logarithm Problem (ECDLP). There is no subexponential-time algorithm is known to solve the ECDLP in a properly selected elliptic curve group. Elliptic curve cryptography offers two major benefits over RSA, it has more security per bit and a suitable key size for hardware and modern communication[1]. Accordingly this results into smaller public key certificates, lower power requirements and smaller hardware processors [2].

2 PURPOSE OF THE SITUATION

In order to secure messages, there are mathematical techniques that provide security services such as confidentiality, integrity, authentication and non-repudiation. [3] , [4], [5].

Confidentiality: data is kept privately in an electronic communication. It is typically provided by encryption. It contains both protections of the transmitted data between two ends. It similarly secures the traffic flow analysis.

Integrity: data is not changed in an unauthorized manner. It is typically provided by digital signature and encryption as well.

Authentication: receiver determines its source to confirm the sender's identity by using something that you have or you know. Normally, it is done by using the sender public key. It is the same integrity provided by digital signature.

• *Corresponding author: Hero Modares is currently pursuing PHD candidate in computer science in University of Mlaya, Malaysia, PH-0060176664612. E-mail: Hero.Modares@IEEE.org*

Non-repudiation: It ensures the sender and receiver from denying the sending or receiving of a message and the authenticity of their signature. Typically, it is provided by digital signature [6]

A digital signature is an electronic signature that can be used to authenticate the identity (Authentication) of the sender of a message or the signer of a document, and possibly to ensure that the original content of the message or document that has been sent is unchanged. Digital signatures are easily transportable, cannot be imitated by someone. The ability to ensure that the original signed message arrived means that the sender cannot easily repudiate it later (Non-repudiation).

A digital signature can be used with any kind of message, whether it is encrypted or not, so that the receiver can be sure of the sender's identity and that the message arrived integral. A digital certificate contains the digital signature of the certificate-issuing authority so that anyone can verify that the certificate is real.

3TARGET READER

Digital signature will dramatically alter the way the world communicates. Essentially, this technology will allow us to conduct legally binding paperless communication. With digital signatures, we can virtually throw away our original to follow by mail stamps and carry out immediately communications and commerce around the world.

Digital signatures can be used for almost any transaction that currently requires a signature. Probable uses include anything from online college applications to the filing of state income tax forms to applications for business permits at the local level. Almost any transaction that requires a signature can be present electronically with the digital signature technology.

In California law, a digital signature is defined as an electronic identifier, created by computer, intended by the party using it to have the same force and effect as the use of a manual signature.

4 FUNCTIONALITY OF SYSTEM

Assume Bob were going to send the text to him lawyer in another town. He wants to assure the lawyer that data was unchanged which was sent from him (Figure 1).

1. Copy the text into an e-mail note.
2. Using special software, Bob gets a message hash (mathematical summary) of the text.
3. Bob uses a private key that he has previously got from a public private key authority to encrypt the hash.
4. The encrypted hash becomes him digital signature of the message.

At the other end, him lawyer receives the message.

To make sure it's intact and from him,

1. Lawyer makes a hash of the received message.
2. Then uses his public key to decrypt the message hash.
3. If the hashes match, the received message is valid.

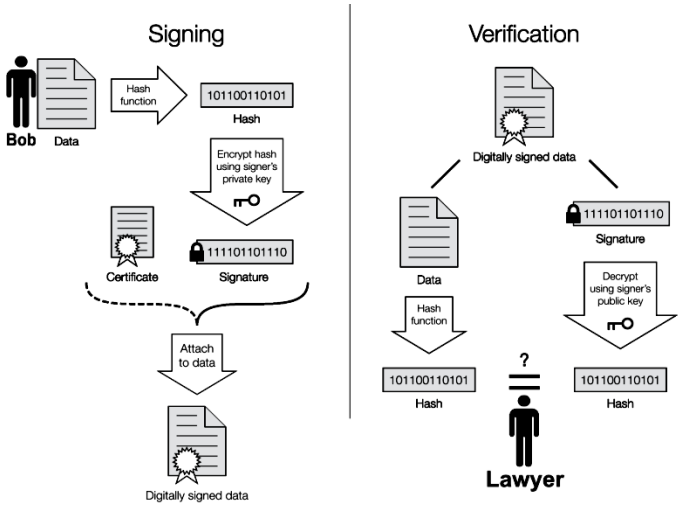

Fig.1 Signing and verification.

5 JUSTIFICATION OF PROBLEMS

Some of the problems are as follow:

1. Companies often possess data files on employees which are confidential, such as medical records, salary records, etc. Employees will feel safer knowing that these files are encrypted and digitally signed and are not accessible to casual inspection by data entry clerks (who may be bribed to obtain information on someone).
2. A company may wish to transfer sensitive business information between sites such as branch offices. Or it may wish to send confidential information (for example, a negotiating position, operating procedures or proprietary data) to an agent in the field (perhaps abroad). If the information is encrypted and digitally signed before transmission then one does not have to worry about it being intercepted since if this happens the encrypted data is incomprehensible (without the encryption key) and digital signature will make sure that it comes from the authenticated user.
3. A company may have information that a competitor would like to see, such as information concerning legal or financial problems, results of research, who the customers are and what they are buying, information revealing violations of government regulations, secret formulas or details of manufacturing processes, plans for future expansion or for the development of new products.
4. Two individuals may wish to correspond by email on matters that they wish to keep private and be sure that no-one else is reading their mail.

From the above examples it can be seen that there are two general cases when digital signature and encryption is needed:

(a) When information, once digitally signed and encrypted, is simply to be stored on-site (and invulnerable to unauthorized access) until there is a need to access that information.

(b) When information is to be transmitted somewhere and it is digitally signed and encrypted so that if it is intercepted before reaching its intended destination the interceptor will not find anything they can make sense of or even if the data is

tempered or not sent by the original user it can be tell by digital signature.

Elliptic curve cryptography is a public-key cryptosystem that is evolving as an attractive option to other schemes such as DSA and RSA. ECC has many advantages over RSA such as the smaller keys that are provided by ECC as it is shown in Table 1. NIST Recommended Key Sizes. In addition, ECC will works on a small parameters size corresponds to the used key. Small parameters size that is provided by ECC made it possible to implement public key infrastructure in constrained application such as RFID and wireless sensor networks. Elliptic curves are using Secure Random method to produce seeds which used in generating the curve parameters.

Many problems were faced during understanding the elliptic curve cryptography mathematics in this work. The lack of the examples and the algorithms explanations made it uneasy to follow these operations in this field level.

Whenever the Received Signal strength (RSS) from the associated Access Point (AP) gets weaker, the Mobile Station (MS) starts the discovery phase by switching to each channel defined by the standard used (11 channels in IEEE802.11b/g and 32 channels in IEEE802.11a) and scans for any available APs [2]. The MS does this because it does not have any information about the surrounding APs so it must discover them itself. There are two kinds of scanning defined by the IEEE802.11 standard, active and passive scanning. In active scanning, the MS scans each channel by sending probe request frames and waits for responses from all available APs on that channel. This scanning type can take a long time, up to 400ms, as the MS must wait for the MinChannelTime (minimum channel timer is used by the MS to specify the minimum time needed for receiving responses from APs) on each channel while it is being scanned [4]. On the other hand, by using passive scanning, the MS only switches on each channel and waits for beacons sent by the APs located on that channel. Although this type looks easier done by MS as it does not consume a lot of power or bandwidth, it takes longer than active scanning because MS has to wait for at least a one-beacon interval on each channel (normally 100ms) introducing big delays (~1s) which are not acceptable in real time applications.

6 Elliptic Curve Cryptography (ECC) Background

Elliptic Curves have been studied by mathematicians for more than a century. One example is proving Fermat's Last Theorem which says that the equation xn +yn = zn has no nonzero integer solutions for x, y and z, in case that integer n is larger than 2. Elliptic curve was introduces in cryptography field in twenty four years ago. It is a quite new cryptosystem, suggested separately in 1985 by Miller [7] and Koblitz[8].

Nowadays, ECC has been adopted by many standardizing bodies such as ANSI [9], IEEE [10], ISO [11] and NIST [12].

Elliptic curve cryptography system is based on Discrete Logarithm Problem (DLP). A group structure that is provided by elliptic curves defined over a finite field is used to implement the cryptographic schemes. The elements of the group are the rational points on the elliptic curve, together with a special point $\mathcal{O}$ which called point at infinity.

The group operation is the addition of points, which can be carried out by means of arithmetic operations in the underlying finite field. Scalar point multiplication is one of the major buildings of ECC block. An operation of form $k.P$ where k is a positive integer, P is a point on the curve. The idea is adding the point P to itself $k-1$ times to get the resulted point$Q$. To recover k from Q and P is known as Elliptic Curve Discrete Logarithm Problem (ECDLP). Until now there is no subexponential-time known algorithm to solve ECDLP in a properly selected elliptic curve group [13]. The best algorithm for the ECDLP is fully exponential time.

Hence, cryptosystems that are relying on the ECDLP provide higher strength-per-bit than the other cryptosystems that rely on the IFP or traditional DLP. This makes ECC works on a smaller key sizes and less memory demands than other traditional DLP-based schemes. Moreover, the gap between ECC and its competitors in terms of key size required for a given security becomes significantly more distinct, e.g., 256-bit instead of 3072-bit (1:12) key size ratio as it is shown in table 3-1.

The US Government National Security Agency (NSA) announced the Suite B cryptographic which is built on ECC specification for sensitive but unclassified communications. Also many companies purchased license to use ECC in their products (Certicom, 2001).

Appraisals are given for parameter sizes which provide a comparable security levels for DL, RSA, and EC systems. Assuming that the mentioned algorithms are the most significant one in it is mathematical problem (ECDLP, DL, IFP).

TABLE 1. NIST RECOMMENDED KEY SIZES

shared Key Size(bits)	EC Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	key Size RATIO
80	163	1024	1:6
128	256	3072	1:12
192	384	7680	1:20
256	512	15360	1:30

The key sizes which provide the same security levels for DL, RSA and EC systems as 80, 128, 192 and 256-bit shared-key encryption scheme are listed in Table 1. NIST Recommended Key Sizes 3-1. The comparisons in Table 3-1 show that smaller parameters can be used in elliptic curve cryptography (ECC) than that in DL and RSA systems at a specific security level. The difference in parameter sizes shows the difference in the security level, the bigger is the parameter, the higher the security level. The benefits that can be obtained from smaller parameters comprise speed and smaller keys and certificates. Especially, private-key operations for ECC are more efficient

than that in RSA and DL private-key operations as well the public key operation such as signature verification ECC are many times more efficient than for DL systems.
Elliptic curve applications are by default using secure random generator to generate the seed which will be used to produce either the curve or the privet key. This generated random number is not fully secure where cryptanalysts may exploit it [14].

6.1 Elliptic Curve Cryptography Domain Parameters

Several parameters define the Elliptic Curve. The elliptic curve points also are characteristic of a particular elliptic curve. Federal standards decided what the domain parameters for particular elliptic curves. Those standards simplified the implementation of EC standards for cryptography and signatures (ECDSA). In addition to the curve parameters a and b, there are other parameters that communicating parties must agreed on them. These parameters are the domain parameters. The domain parameters for both prime field and binary field are given below:
D = (q, FR, a, b, G, n, h), where
q: is the field size (q=p or $q=2^m$) p is prime.
FR: shows the method that is being used to represent the elements over the curve
a,b : The parameters that defining the curve
G: The base point or the generator point
n: The order of the base point. Such that n.G=$\mathcal{O}$
h: cofactor, where h= $\#E(\mathbb{F}_q)/n$, where $\#E(\mathbb{F}_q)$ is the number of points on the curve.

6.2 Elliptic Curve Discrete Logarithm Problem

After going through the basic mathematics that is used with elliptic curve cryptography, I can go through the elliptic curve discrete logarithm problem (ECDLP), elliptic curve cryptography basis. Fix elliptic curve E, xP represents the point P added to itself x times. Suppose Q is the resulted point from adding the point P to itself x times or Q=xP.
Then the Elliptic Curve Discrete Logarithm Problem is to determine x given P and Q. ECC security emerges from the difficulty of resolving the Elliptic Curve Discrete Logarithm Problem ECDLP. As the case for Integer Factorization problem and the discrete Logarithm problem modulo p, there is no efficient method to solve ECDLP at this time [15].
ECDLP is believed to be harder than the Integer Factorization problem and the Discrete problem, which implies that ECC is one of the strongest public key cryptography available today [15].
Elliptic curve cryptography is built on similar ideas to the ones used for discrete logarithm systems (DL), the different is that the DL functions are performed on elliptic curves over finite fields [16].
The important feature in accepting ECC is the smaller cryptographic key sizes [16]. With small key sizes public key cryptography can be used on constrained applications such as RFID and wireless sensor networks.

6.3 special classes of elliptic curves

There are special classes of the elliptic curves that are vulnerable to particular attacks. These classes are detailed below [16].

6.3.1 Supersingular curves

The curve $E(\mathbb{F}_q)$ is said to be supersingular if it is of field of characteristic two and has j-invarient equal to zero [17]. j-invarient is defined in chapter 4.
Menezes[18], Okamoto, and Vanstone [19] showed that ECDLP in an elliptic curve E defined over a finite field F_q can be reduced to the DLP in some extension field. It is practical if the exponent of the extension field is small. To ensure that this reduction method does not apply to the curve, the order of the point does not divide $q^B - 1(1 \leq B \leq 2000/(\log_2 q))$.

6.3.2 Anomalous curves

An anomalous elliptic curve over Fq is an elliptic curve which has precisely q points. By verifying that the number of points on an elliptic curve does not equal the number of elements in the underlying field, easily can be ensured that the Smart-Satoh-Araki attack does not apply to the curve - these curves are specially prohibited in all standards of elliptic curve systems.

6.4 Elliptic curve digital signature algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the DSA. ECDSA was first proposed in 1992 by Scott Vanstone [20]. ECDSA was first proposed in 1992 by Scott Vanstone [21] in response to NIST (National Institute of Standards and Technology) request for public notes on their proposal for DSS. In 1998 the International Standards Organization (ISO) accepted this proposal, after that by one year the American National Standards Institute (ANSI) accepted the proposal to and issued the standard ANSI X9.62. IEEE (Institute of Electrical and Electronics Engineers) accepted as standard (IEEE 1363-2000). The higher level in my design is ECDSA. As the highest level it will use the lower levels functions and algorithms to generate and to verify the signatures.

6.4.1 ECDSA key pairs

As it is mentioned earlier the domain parameters that the communicating parties must agree on the domain parameters. These domain parameters will be the public key. Whereas, the private key is a generated random number and it is used to generate the multiple.

6.4.2 ECDSA signature generation and verification

This sector shows the procedures for generating and verifying signatures using the Elliptic Curve Digital Signature Algorithm ECDSA.

6.4.2.1 ECDSA signature generation

Let us assume that Alice has the domain parameters D=(q, FR ,a, b, G, n, h) and pair of keys (d,Q) she can does the following to sign the message m.

1. generate a random number k , where $1 \leq k \geq n-1$
2. computekG=(x_1, y_1), convert the value of $x_1$to integer $\bar{x}_1$
3. Calculate the arithmetic modulo of $\bar{x}_1$, $r = \bar{x}_1 \bmod n$. If r=0 then return to step 1.
4. Calculate $k^{-1} \bmod n$
5. Compute the e=SHA-1(m).
6. Calculate the equation $s=k^{-1}(e+dr) \bmod n$. If s=0 then go to the step 1.
7. Alice signature for the message m is the pair numbers (r,s).

6.4.2.2 ECDSA signature verification

For Bob to verify the signature (r,s), Bob get a copy of Alice domain parameters D=(q, FR ,a, b, G, n, h) as well as her public key. To verify the signature Bob does the following:
1. Compute e=SHA-1(m)
2. Compute $w = s^{-1} \bmod n$
3. Compute $u_1 = ew \bmod n$, and $u_2 = rw \bmod n$
4. Compute the resulted point $Y=u_1G + u_2Q$, Q is the public key
5. If $Y=\mathcal{O}$ then reject the signature. otherwise calculate $v=\bar{x}_1 \bmod n$
6. The signature is accepted if and only if v=r, otherwise reject the signature.

The implementation in the protocol level is working on ECDSA, so all the generation and verification is using the ECDSA algorithms.

7 HASH

Hashing algorithm is used to insure the integrity of the received data and is used to detect any single-bit errors. The one-way hashing algorithm takes variable-length of data even thousands or millions of bits and produces a fixed-length output. The hash function ensures the stability of the data from changing as it may produce a completely different output value [22].

Currently, there are currently several different hashing algorithms such as LM, MD5, SHA-1, and SHA-2. They can be used for many different aims like digital signatures, data fingerprinting and message authentication codes. The main idea of hashing is to provide confidence and security of the authentic data transmission to user. Cryptographers and hackers always attempt to find subject of each successive algorithm that is released. If one algorithm is broken, another algorithm is developed and the cycle continues [22].

7.1Hash Algorithms

There are fourteen different types of hash algorithms used today [23]. There is a discussion about cryptographic functions of LM (LanManager), MD5, SHA-0/SHA-1, and SHA-2. SHA-0 and SHA-1 are closely familiar and they are commonly approached as a single algorithm [22].

7.2 Uses of Hashing

Hash algorithms were used to detect errors in a file's transmission. However, presently Hash algorithms have more functions. Therefore, it needs to be more secure. One of its abilities is to reduce the data to a manageable signature element in digital signature. After reducing the data, signature element is sent to authority such as RSA or ECC to be digitally signed. [24]

Another ability is Message Authentication Codes (MAC) which is one-way hash algorithm with the addition of a secret key [25]. Secret key is the only way to verify a MAC value and the hashing. The use of the keyhappens in multiple times in some MAC algorithm [24].

Hashing is as a Pseudo-random function. By inputting data through a hashing algorithm, it results in random-seeming bits. Use Diffie-Hellman to exchanges these random-seeming bits to be used later for generating cipher keying material [24].

Data fingerprinting is the fourth use of hashing and probably the most common use of hashing. The output identifier can be used to verify integrity of file or a message. Even there is a slight modification in the file; the resulting hash of the file is totally different. Hashing algorithm input data can change up to 50% of the produced hash even if it is single bit.

8 DIAGRAM

8.1 Use Case Diagram

The Use Case Diagram is used to capture the relationship between actors and the functionalities of the system. The formal definition for a use case is: A use case is a sequence of transaction performed by a system that yields a measurable result of values for a particular actor [26].

In terms of a Safe Message system there only are two actors; one actor sends a message to another actor. Both actor are given the own set of public and private keys(Figure 2).

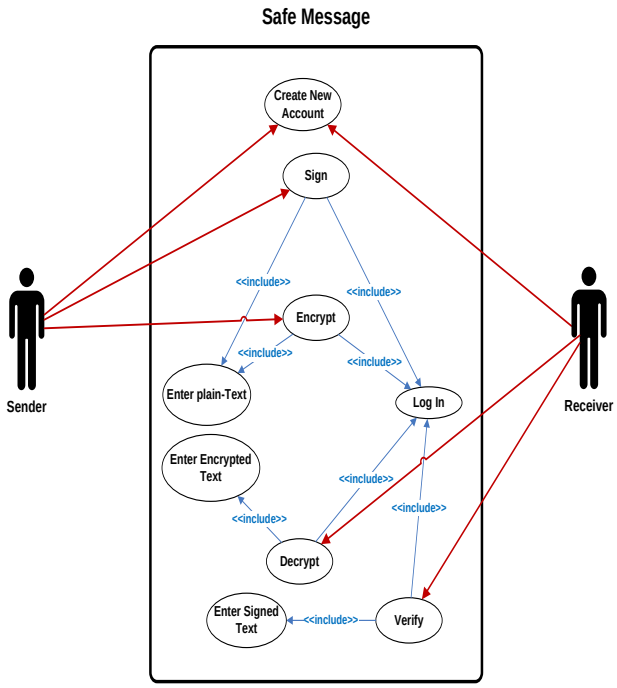

Fig.2 Use Case Diagram.

8.2 Data flow diagram

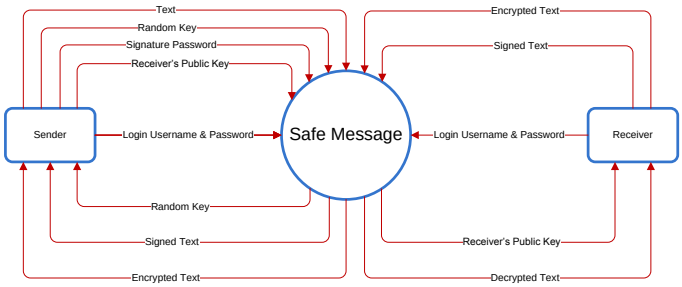

Fig. 3 Data flow diagram - Leve0 diagram.

8.3Sequence Diagram

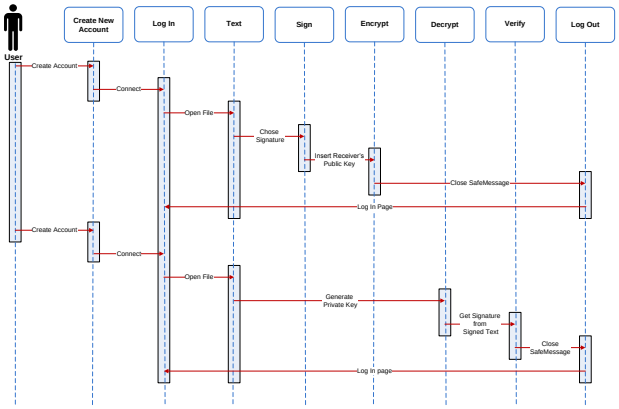

Fig. 4 Sequence diagram

8.4Activity Diagram

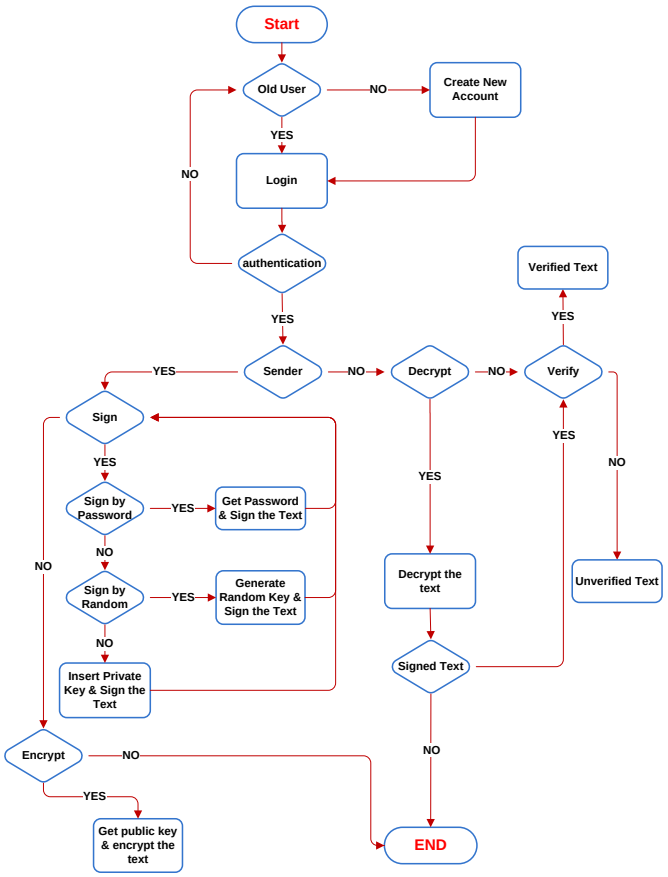

Fig. 5 Activity diagram.

After coding the classes, reasonably it has to be integrated to an appropriate interface so that users can perform and react to the actions of the system. Users' requests are sent as messages to a particular function or class for processing, and the results are shown on the interface. In addition, it is designed to prevent the user from entering invalid inputs which may corrupt the results.

9 TESTING

Testing is a very important phase in the development of any system. A test is done to evaluate the quality of the system and the degree of success as well as to check if the system has met the project specification. A test plan is drown up to facilitate and create a systematic approach efficiently test system. In most recent development methodology, test plan are created even before the construction phase. The purpose of testing has become so essential to the system development that modern development methodologies encourages testing to be carried out from time to time during the whole system development life cycle. This incorporate modular testing in modern day test plans.

However, to evaluate the actual quality of the system a complete version of the system is required. Here many criterions are tested base on the deferent types of testing. It ranges from functional testing to user acceptance testing. The

main purpose of testing is to identify problems that might occur and to measure system performance. In this section, we shall discuss the various types of testing that were drawn out to evaluate the system's performance and success.
There are four major testing plan has been taken in this project, which included module testing, integrity testing, system testing and user acceptance testing.

9.1 Module testing

Module testing is a collection of dependent components that encapsulate related components only. Therefore, it enables each module to be tested independently. This testing will ensure that the module calling sequence in this project is systematic. The mine purpose of the test is to verify the correctness of the flows of events. Therefore, with the system development process being carried out module by module, the module testing will also be carried out once a module has been completed.
All modules in Safe Message have been tested to verify their degree of achievement in this project and expected results arrived in an accurate manner. This test is conducted after implementation all the functionalities into the system. This is to ensure that the entire modules are working properly.

9.2 Integration testing

Integrity testing is the process of verifying that the system component will work together as describe in the system and program design specification. In this phase the test is conducted on the interface of two interactive components in a single unit. This involves the examination process of interfaces in the system and in continuo unit the entire system is developed.
In general, integration testing is carrying out to ensure the interface between modules can function properly. The most common problem that occurs in large software system is subsystem interface mismatches. The subsystem test procedures should concentrate on the detection of interface error by vigorously exercising those interfaces.
The communication between login page and main page is the main focus of this testing. Multiple testing was carried out to ensure the handshaking within modules and hardware is in proper manner.

9.3 System testing

Final testing procedure done is system testing. However, testing the system at whole is very different from previous module testing and integration testing. System testing is a series of different tests designed to fully exercise the software system to uncover is limitation and measure its capabilities. The objective is to test an integrated system and verify that it meets specified requirements. Although each test in this project has a different, all work to verify that system elements have been properly integrated and perform allocated functions.

9.3.1 Function testing

System testing exists with function testing, which is based on the system's function requirements Function testing is performed in a carefully controlled situation. Function testing is based on the system function requirements in order words, a function test is used to check that whether the integrated system performs its functions as specified in the requirements.

9.3.2 Performance testing

Performance testing addresses the non functional requirement of the system after function testing is completed. System performance is measured using performance objective set by potential users as highlighted in the non functional requirement section as guideline. The purpose of this system is to test run-time performance of software within the context of an integrated system. It requires both hardware and software instrumentation.

9.3.3 Robustness testing

Java is generally known as one of the most robust programming language available today. This is due to its exception handling which implements try, throwing and catch command that give room for error handling. This system has implemented in the system to ensure robustness of the system.

9.3.4 User acceptance testing

This testing was conducted during and after system implementation phase. User acceptance testing is the best technique to discover the overall system operations. It provides user the opportunity to tell whether the system meets their requirements. The degree of satisfaction using the system and the user feedback was determined.
In order to complete the testing stage a full working system is put to the test is the end users hands.

10 CONCLUSION

Nowadays, RSA generally uses public key cryptosystem in most applications that use PKC. However, recently ECC has a trend which makes it become the convenient cryptography system. ECC is also becomes substitute for RSA in efficacious applications caused by its efficiency in software as well as in hardware realizations. ECC provides a better security with shorter bit sizes than in RSA. Shorter key length saves bandwidth, power, and it enhances the performance. In contrast with the past, pairing in ECC attracts more attention of experts because it can be used to build a number of cryptographic schemes that cannot be constructed in any other way. The research starts with survey of cryptography, Elliptic Curve arithmetic and Elliptic Curve operations hierarchy algorithms. A code is written to generate and verify digital signature ECDSA. SHA-1 is chosen as a hashing algorithm as it is recommended by ANSI. The code is written in four phases, which are generating the signature, verifying the signature, encrypting and decrypting.

The author was able to demonstrate the Elliptic Curve Digital Signature (ECDSA) using the polynomial representation over the binary field. The establishment of the custom parameters requires a thorough understanding of the application and the algorithms which are used in generating the parameters.

An important feature of cryptosystems is that they are relying on the Elliptic Curve Discrete Logarithm Problem (ECDLP), and the problem is that they provide a higher strength per-bit than other cryptosystems which rely on different mathematical problems. This feature permits the ECC to use smaller key sizes which demand for less memory than that in the other traditional DLP-based schemes. Other benefits which can be obtained from smaller keys are speed and smaller certificates.

It is important to highlight that the objectives outlined in this work have been achieved by implementing the Elliptic Curve Digital Signature Algorithm (ECDSA). In addition, the researcher has also managed to use the multi ways to generating both the elliptic curve domain parameters and private keys and use them in generating the ECDSA signature. In addition, the system is limited to curves with the size of 162 bits.

ACKNOWLEDGMENTS

This work was supported in part by the University of Malaya, Kuala Lumpur Malaysia under UMRG Grant (RG080/11ICT).

REFERENCES

[1]T Kerins, E Popovici, WP Marnane, and P Fitzpatrick, "Fully Parameterizable Elliptic Curve Cryptography Processor over GF (2 m)," in 12th International Conference on Field-Programmable Logic and Applications, 2002, pp. 750 - 759.

[2]Hero Modares, Yasser Salem, MohamadRezaHosseiniFatemi, and RosliSalleh, "Application of Elliptic Curve Cryptography," 3rd International Conference on Informatics and Technology 2009, Kuala Lumpur, Malaysia 2009.

[3]G. Stoneburner, Underlying technical models for information technology security.: Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-33, 2001.

[4]I. Riedel, "Security in Ad-hoc Networks: Protocols and Elliptic Curve Cryptography , on an Embedded Platform," Ruhr-Universit□at Bochum. , March 2003.

[5]Hero Modares,., 2009.

[6]Ahmad Khaled M. Al-Kayali, "Elliptic Curve Cryptography and Smart Cards," SANS Institute, p. 2004, February 2004.

[7]V. S. Miller, "Use of elliptic curves in cryptography," Advances in cryptology-- CRYPTO '85, pp. 417-426, 1986.

[8]N. Koblitz, "Elliptic Curve Cryptosystems," Mathematics of Computation, pp. 203-209, 1987.

[9]ANSIX9.62, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)," New York, USA, 1999.

[10]"IEEE P1363-2000: IEEE Standard Specifications for Public-Key Cryptography," MD, USA, 2000.

[11]"ISO/IEC 15946: Information Technology -Security Techniques: Cryptographic Techniques based on Elliptic Curves," Switzerland, 2002.

[12]NIST, "FIPS 186-2:Digital Signature Standard (DSS)," Gaithersburg, 2000.

[13]A. M. Odlyzko, "Discrete logarithms: the past and the future," Designs, Codes,and Cryptography, vol. 19, no. 2-3, pp. 129-145, March 2000.

[14]Jonathan Knudsen, Java cryptography.: O'REILLY, 1998.

[15]Randall K. Nichols, ICSA Guide to Cryptography.: Computing McGraw-Hill, 1999.

[16]William J. Caelli, Edward P. Dawson, and Scott A. Rea, ""PKI, elliptic curve cryptography"," Computers & Security, pp. 47-66, 1999.

[17]A.J. Menezes and Scott A. Vanstone, "Elliptic curve cryptosystems and their implementation.," Journal of Cryptology, pp. 209-224, 1993.

[18]A.J. Menezes, "Elliptic Curve Public Key Cryptosystems".: Kluwer Academic Publishers, 1993.

[19]A. Menezes, T. Okamoto, and S. Vanstone, ""Reducing elliptic curve logarithms to logarithms in a finite field"," in IEEE Transaction on Information Theory, 1993, pp. 1639-1646.

[20]D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," A Certicom Whitepaper, 2001.

[21]S. Vanstone, Responses to NIST's proposal. Commun ACM, pp. 50–52, 1992.

[22]K. C. Redmon, "C0D3 CR4CK3D: Means and Methods to Compromise Common Hash Algorithms," July 2006.

[23]X. Wang, Y Lisa Yin, and H Yu, "Finding Collisions in the Full SHA-1".

[24]S. Bellovin and E. Rescorla, "Deploying a New Hash Algorithm," In a presentation delivered at the Rump Session of CRYPTO 2005, 2005.

[25]B. Schneier, "Applied Cryptography," New York: Wiley Publishing, pp. pp.30-31, 428-459, 1996.

[26]Terry Quatrani, Visual Modeling with Rational Rose 2000 and UML.: Addison Wesley, 2001.