IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Secure Digital Signature with Elliptic Curve Cryptography Scheme using Galois Field

Mohammad Amjad

Department of Computer Engineering Faculty of Engineering & Technology, Jamia Millia Islamia New Delhi, 110025, India

*Corresponding Author: mamjad@jmi.ac.in

Available online at: www.ijcseonline.org

Accepted: 12/Aug/2018, Published: 31/Aug/2018

Abstract-- The development and growth of Internet technology has made digital signature in the convenient and helpful manner for electronic transaction security and business applications. Digital signature is largely used because of its data integrity, protecting the data, privacy and authenticity assets. Digital signatures are mostly used in financial transaction, credential identify and software distribution, where it is indispensable to detect forgery or tampering of data. Digital signatures based on elliptic curves are more secure, reliable and suitable for constrained environments like wireless sensor networks because of its reduced processing overhead. This paper discusses the principles of Digital Signatures, their applications based on Elliptic Curve Digital Signature (ECDSA) scheme using asymmetrical cryptography along with Galois Field. Finally, a practical elliptic curve digital signature system using Galois Field is implemented and its time complexity is analyzed. The time complexity results validation claim that the proposed ECDSA is suitable for use in real time environments like WSN and smart cards. The proposed technique is based on mathematical model used in ECC with Galois Field along with the secure hash function. Programming language Python is used to realize the algorithm used.

Keywords: Digital Signatures, DSA, ECC, Galois Field, ECDSA, Encryption using asymmetric cryptography, Hash Functions, SHA 512.

Nomenclature: DSA (Digital Signature Algorithm), ECDSA (Elliptical Curve Digital Signature Algorithm), SHA (Secure Hash Algorithm), GF (Galois Field).

I INTRODUCTION

With the advancement of technology, the traditional ink signature is no longer reliable or sufficient, leading to the emergence of a new technique based on asymmetric cryptography called digital signature or electronic signatures [3]. Electronic business and communication mediums have a lot of security flaws and loopholes thereby demanding reliability and high security. The data transferred in these applications is usually through an insecure channel. Data however, to remain valid must have the characteristics of authenticity, secure and data integrity [1].Through data integrity protecting and privacy properties digital signature are able to solve this problem. A digital signature is a onetime generated checksum which is a function of input message and the time period during which the signature was generated. A digital signature is a mathematical concept that is computed by employing fixed set of parameters such as the identity of the signatory and pre decided rules that allow verifying the integrity of the data [6][7]. Real number arithmetic is slow in processing as well it doesn't give the precise result because of its approximation value. Cryptographic applications require accurate result giving arithmetic and should be fast in processing also. The Galois Field is one of the component of application of modular arithmetic used for calculate the integers in the given certain range. We use the same addition operation with the calculation done in modular p in the form of E_p (a, b), where p is defined as modulus and a and b are coefficient of the equation $y^2 = x^3 + ax + b$. For example $y^2 \bmod p = x^3 + ax + b \bmod p$ has an underlying field of GF (p) if a and b are in GF (p). For example consider an elliptical curve over the field E_{23} (a, b) with a= 1 and b = 0, the elliptical curve equation will become $y^2 = x^3 + x$. We can find out the point lies on the elliptical curve E_{23} (a, b). For the equation $y^2 = x^3 + a$ on E_{23} (a, b), the point (9 , 5) satisfies this equation since $y^2 \bmod p = (x^3 + x) \bmod p$ so that $5^2 \bmod 23 = (9^3 + 9) \bmod 23$, $25 \bmod 23 = (729 + 9) \bmod 23$, $25 \bmod 23 = 738 \bmod 23$, 2 = 2, This means 23 points are there on the elliptical curve E_{23} (a, b) with a = 1 and b = 0.

The rest of the research paper is organized as follows: Section II is about the related works. Section III highlights the Digital Signature Technology pertaining to this work. Principle and practicality of Digital signature is discussed in Section IV. Introduction to Elliptical curve cryptography is discussed in V. Proposed Elliptical Curve cryptography using Galois field is explained in section VI. Implementation and result is shown in section VII and finally contribution is summarized and concluded in Section VIII.

II RELATED WORK

Elliptic curve cryptography was introduced by Victor Miller and Neal Koblitz in 1985[13]. Elliptic curve cryptography proposed as an alternative to established public-key system such as RSA. This is the type of public key cryptography base on the structure of elliptical curves. This is analogous of existing public key cryptosystems in which modular arithmetic is replaced by operations defined over elliptical curves.

Whitfield Diffie and Martin Hellman in the year 1976 for the first time described the concept of a digital signature scheme, inferring that such schemes existed [14][15]. In 1977, R.L. Rivest, Shamir, and Adleman publicly described the popular RSA algorithm [12]. RSA algorithm is now used to produce primitive digital signatures and encrypt or decrypt the data being sent. Lamport signatures, Merkle signatures and Rabin signatures were soon developed after RSA [4]. The procedure of digital signature requires three steps viz. Key generation, signing algorithm and verification procedure.

III. DIGITAL SIGNATURE TECHNOLOGY

Digital signature requires two sets of the keys, one is the public key and other is the set of private key. Public key cryptographic system involves two mathematically linked distinct keys, public key and private or secret key. The public key is in open domain whereas the private key is known only to the intended recipient of the message. Public key helps to encrypt the message or verify the authenticity of a signed message. Private Key performs the reverse operation. Cipher text is decrypted to original data using the private key. Private Key is also used to generate the digital checksum. The term asymmetric points to the use of separate keys that performs inverse functions. A key aspect of the public key encryption is that it mathematically links the public and private keys so that only a fixed pair of key can be used for encryption and decryption. The strength of the system lies in the fact that it computationally impractical to evaluate the carefully generated private key from its corresponding public key. An example of popular public key encryption program is Pretty Good Privacy (PGP) used in email.

A. Digital signature function:

1. Guaranteeing data integrity: Even if the message is changed slightly by an unauthorized third party, the digital signature becomes invalid and the message is rendered useless [8].
2. Anti-deniability: Public key cryptography algorithms restrict the signatory from wrongfully claiming that he has not signed and sent the received message [11].
3. Anti-fabrication: Prevents the receiver from forging or altering the original message that is claimed to have been sent by the signee [12].

B. Secure Hash Functions

Cryptographic hash functions (Eq.1) are considered practically impossible to invert which means it is not feasible to deduce the input data given only its hash value. Hash functions are capable of converting the input of varying lengths to fixed length outputs.

The input to these one way hash function is the message or document, and the output is the message digest or hash value. Mathematically a hash function is defined as

$$F:\{0, 1\}^* \rightarrow \{0, 1\} \quad \text{(Eq. 1)}$$

Where * indicates that for any large pattern of the repeated sequence it produces fixed amount of output always. A hash function is cryptographic secure, if it is:

1. Pre-image resistant: Hard to find any message m for which the hash value h = F (m) i.e. Hash is one way function.
2. Second pre-image resistance: Hard to find for a given m_1 and any other m_2 with the same hash value i.e. F (m_1) ≠ F (m_2).
3. Collision resistant: Hard to find any pair of m_1 and m_2 with F (m_1) = F (m_2).

C. Attacks on Digital Signature

1. Key only attack: In this attack, the attacker is given only the public verification key of user A. To forge data, the attacker must create A's signature and thus successfully create an illusion that the message is sent by the sender A. This is same as the cipher text attack.
2. Known message attack: In this type of attack an attacker has access to some of the valid signatures generated by user A on messages known to the attacker but not chosen by him. Attacker then creates another message and tries to forge A's signature on it. This is similar to the known plain text attack (KPA).
3. Chosen message attack: In this attack, the attacker tries to learn A's signature by making him sign on some messages chosen by the attacker. Now the attacker creates another message containing the data that he wants and tries to forge A's signature on it. This is similar to the chosen plain text attack (CPA).

D. **Practicality of Digital Signatures**

1. Authentication: This refers to the ability to positively confirm the truth of an attribute of a specific piece of data claimed true by another entity. It helps to authenticate an individual's identity involved in an electronic transaction. Since a digital certificate is proofed by a trusted third party, it undisputedly identifies a person with its true identity.
2. Data integrity: This refers to the fact that data is in its original form and is not tempered by an unauthorized entity. Mostly during electronic transactions, data flows through insecure networks channels. This makes it essential to ensure that data remains intact, and is not altered in any way while it travels from its source to the intended destination. Digital signatures are glued to the data in such a way that any third party change in the data content will remove this binding thus rendering the signature invalid.
3. Non-repudiation: This refers to assurance that the sender will not deny having signed a message. It binds the signer to the information that they sign. Similar to real-world business process, once you sign a document, you are legally bound to its content.

E. **Applications of Digital Signature**

Digital signatures reliably automate the signatures of authorizations thereby eliminating the need of ink signatures. This makes the process relatively cheap with increased efficiency. Digital signature are particularly useful for:

1. Business: Documents related to sales, proposals, contracts with customers, purchase orders, agreements with clients, leases, reports and reimbursement approvals.
2. E-commerce: Electronic tendering systems and web portals with external modules to sign the online transactions
3. Human Resources: Documentation of employees, securing details of ration-cards, use in wireless communication
4. Government: Electronic data exchange between high officials. In India, the government widely uses digital signature for personal identification like AADHAR, online railway ticket booking and refund.
5. Medicine: Patient consent forms, patient history, medical exams, prescriptions and report analysis.

.

IV. PRINCIPLES OF DIGITAL SIGNATURE

For any of the digital signature generation we require the three steps:

1. A typical key generation method, which is supposed to be the strongest part of the signature generation procedure. This means that key should be strong enough susceptible to any attack.
2. Signature generation by using the set the pair of keys used. Generally set of private key is used for the generation of digital signature.
3. Finally the verification of the generated digital signature and it is verified by using set of public keys.

A. **Key Generation**

1. Choose a prime divisor x which is a 160-bit integer.
2. Find y, a 1024 bit prime number such that $(y-1) \pmod x = 0$. y is called the prime modulus.
3. Choose an integer z in range (1,y), such that $z^x \pmod y = 1$ and $z = h^{((y-1)/x)} \pmod y$, where $h \in (1, y-1)$
4. Select an integer $k \in (0,x)$
5. Determine n as $z*k \pmod y$. (y, x, z, n) and (y, x, z, k) form the public key pair and the private key pair respectively.

B. **Signature Generation**

1. Using an approved cryptographic hash algorithm (SHA-1) determine the message digest H
2. Choose a random per-message value d, such $d \in (0,x)$
3. Calculate g as $(z^d \bmod y) \bmod x$. Repeat step 2 if g=0
4. Calculate $s = d^{-1}*(H(M)+(r*k)) \bmod x$. Repeat step 2 if s = 0.

Signature for the message is (g, s).This is graphically shown in Figure 1.

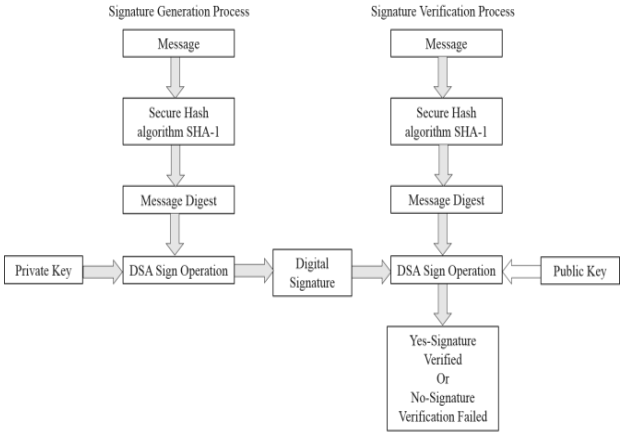


Fig 1: Principle of Digital Signature

C. **Signature Verification**

1. If either $0<r<x$ or $0<s<x$ reject the signature
2. Using the same hash algorithm generate the message digest H of the signature,
3. Calculate $w = s^{-1} \pmod x$.
4. Compute $s_1 = H * w \pmod x$.
5. Evaluate $s_2 = r*w \pmod x$.
6. Determine $v = ((z^{u1})*(n^{u2}) \bmod y) \bmod x$.

The digital signature is valid if v=g.

V. ELLIPTIC CURVE CRYPTOGRAPHY

A. Elliptic Curve Cryptography functions

The use and implementation of elliptic curves in cryptography was independently suggested by Neal Koblitz and Victor Miller in 1985. They are viewed as elliptic curve analogues of the discrete logarithm. They are based on algebraic structure of elliptic curves over a Galois or finite field. The computational intractability of ECDSA which states that given the publicly known base point it is infeasible to determine the discrete logarithm of an elliptic curve element forms the basis of security for ECC.

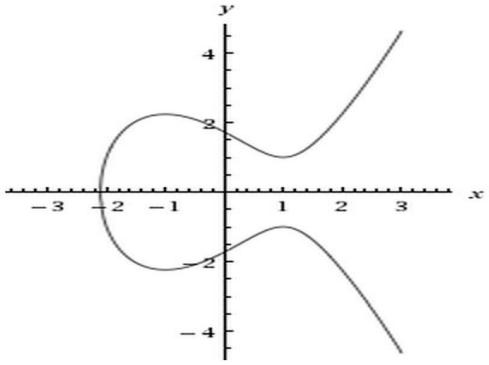


Fig 2: Elliptical curve $y^2= x^3-2x+2$

Using ECC it is easy to compute point multiplication but nearly impossible to determine the original multiplicands given the product.

The following equation describes an elliptic curve E over Z_p in the Cartesian coordinate system:

$$y^2 = x^3 + ax + b \qquad \text{(Eq. 2)}$$

Where a, b $\in Z_p$ and

$$4a^3 + 27b^2 \pmod p \neq 0 \pmod p,$$

A special distinguished point at infinity is denoted as O.(Eq. 2) is commonly referred to as the Weierstrass equation. With a=-2, b=2 the elliptic curve is obtained as shown in Fig. 2.

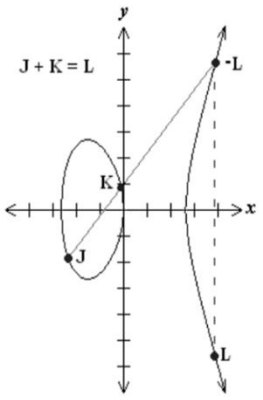
The set E ($Z_{p)}$ consists of all points (x, y) such that both x and y belong to Z_p. These x and y also satisfy the defining curve equation, together with the point at infinity O. The public key is obtained by multiplying the private key with a generator point G on the curve. The public key is a point on curve and the private key is a message specific one time random number. Elliptic curves have application in encryption, digital signatures and pseudo-random generators. Elliptic curve cryptography is vulnerable to side channel attacks and quantum computing attacks [10].

B. Elliptic curve operations over Galois or finite fields

A finite field or Galois field refers to a set of elements. Two basic binary operations namely addition and multiplication are defined on this field that satisfy certain arithmetic properties and rules. The number of elements in the field is called its order. A finite field of order g and denoted by F_g exists iff x is a prime power x^k, where x is a prime and $k\in(0,\infty)$. Also in a field F_g with prime power x^k adding an element p times results in zero; i.e. the characteristic of the field is x. Fields with the same order are called isomorphic [5].

The basic operations are described below:

1. Point addition: It is the summation of two points J and K to obtain a third point L i.e. L= J + K.
   If K≠-J then a line drawn through the points J and K will intersect the elliptic curve at exactly one point i.e. at point –L. The sum of points J and K can be obtained by taking the reflection of the point -L vertically with respect to the X-axis.
   If K = -J the line passing through these points intersect at infinity O. Hence J + (-J) = O. A negative of a point is the replication of that point with respect to X-axis. This is shown in Fig.3.
2. Point doubling: It is the addition of a point J to itself to obtain a new point L i.e. L = 2J.
   If a tangent is drawn at point J it will intersect the elliptic curve at exactly one more point ie. at point –L. If y coordinate of the point J is non zero then the point which is the double of the point –L can be obtained by taking the replication of –L vertically with respect to X-axis. This gives the point L.
3. The tangent at point J will intersects at a point on infinity O if y coordinate of the point J is zero. Hence 2J = O when y_j=0. This is shown graphically in Fig 4.



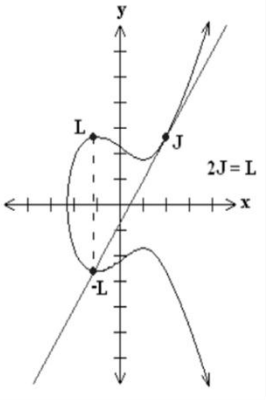


Fig 3: Point Addition Fig 4: Point Doubling

VI. PROPOSED ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM (ECDSA)

ECDSA is the elliptic curve referent of DSA and serves the same purpose of key generation, signature generation, and signature verification. In 1992, ECDSA was publicly proposed by Scott Vanstone of University of Washington. This was in response to National Institute of Standards and

Technology (NIST) request for public advices on their first proposals for digital signature standards [6].

A. Key Generation

Let user A signs a message M. For some prime g and finite field F_g, select an elliptic curve E. Select a point P of order k such that $P \in E (F_g)$. Now A is to generate the public private key pair performs the following steps:

1. Choose a unique and unpredictable integer i such that $i \in [1, k- 1]$.
2. Determine Q = iP.

A's public key is the above calculated Q and private key is i.

B. Signature Generation

To sign a message M with the private key i, user A performs the following steps:

1. Choose a random or pseudo random integer m such that $m \in [1,k-1]$
2. Calculate mP as key pair (x_1, y_1) and $r= x_1 \pmod k$, where x_1 is an integer and lies between 0 and g-1. If r= 0 then repeat step 1.
3. Compute $m^{-1} \pmod k$
4. Calculate H=h(m) where h is the Secure Hash Algorithm (SHA-1)
5. Determine $s= m^{-1} \{H+ i*r\} \bmod k$, If s = 0, then go back to step 1.

The pair (r,s) is the signature for the message M.

C. Signature Verification

To verify A's signature (r, s) on the message M, user B first obtains an authenticated copy of A's public key Q. B now does the following:

1. Verify that r and s belong to the interval [1, k-1].
2. Determine $w = s^{-1} \bmod k$ and H=h(M) where h is the Secure Hash Algorithm (SHA-1)
3. Evaluate $s_1 = H*w \pmod k$ and $s_2 = r*w \pmod k$.
4. Calculate $s_1P + s_2Q = (x_0, y_0)$ and $v= x_0 \pmod k$.

If v=r, the message M is verified for authenticity.

D. Comparison of ECDSA with DSA

1. To provide the same level of security ECC uses 160 bit arithmetic as compared to 1024 bit operations in DSA (ANSI). This also makes ECDSA faster than DSA.
2. ECDSA and DSA algorithms are variants of El-Gamal signature scheme.
3. Both ECDSA and DSA have fast signing and slow verification characteristics.
4. Generating system domain parameters is time consuming step in the both algorithms
5. In their existing versions, both ECDSA and DSA use SHA-1 as the major cryptographic hash function [7,8].

E. Advantages of ECDSA

1. Greater security for a given key size when compared with RSA and other similar algorithm..
2. Effective and easy implementations for cryptographic operations.
3. Less time required to generate and validate signature when compared with its counterparts.
4. Suitable for devices having less computing power and little memory.

VII. MPLEMENTATION AND RESULTS

Table 1: Comparison of key sizes in bits

Digital signature algorithm (DSA)	Elliptic curve Digital signature algorithm(ECDSA)
512	106
768	132
104	70
2048	87
4096	600

For the purpose of comparison between standard digital signature technique and the proposed method, the key sizes of both the methods are compared. The research work is divided into well-defined modules and procedures e.g. elliptical curve parameter generation, key generation, signature generation and verification steps were carried out in separate modules to keep a track of the time for individual operation. Intel Core i5-5200U CPU is used to conduct the experiment at 2.20 GHz clock speed, 1TB hard disk and 4.00GB RAM in the language Python.

For the sake of understanding and analysis of results a simple elliptic curve $E_{997}(1,17)$ is chosen.

$$y^2= ((x^3+x+17) \pmod{997})$$

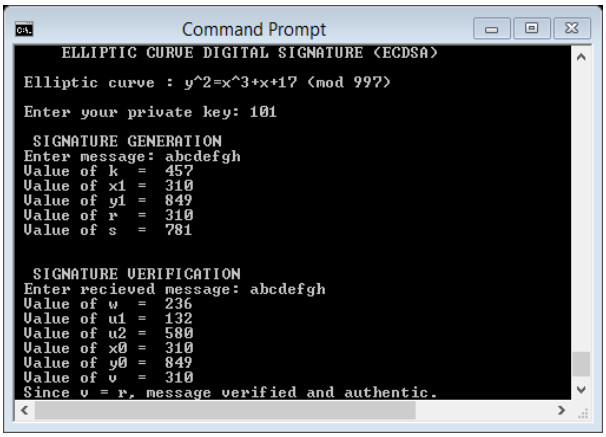


Figure 5: Signature verified V=R for Private key =587

Figure 5 shows the scenario when the original message is same as the received message. ECDSA validates and authenticates the message. This gives sufficient reason for the receiver to believe that the message is authentic and is sent by whom it appears to be sent from. On the other hand, figure 6 shows the situation when the received data is different from the message sent by the signer. Slight tempering and changes with data makes the signature and message invalid. This cautions the receiver against believing in the forgery of the signature and the message. Values of each variable used in the algorithm are shown separately.

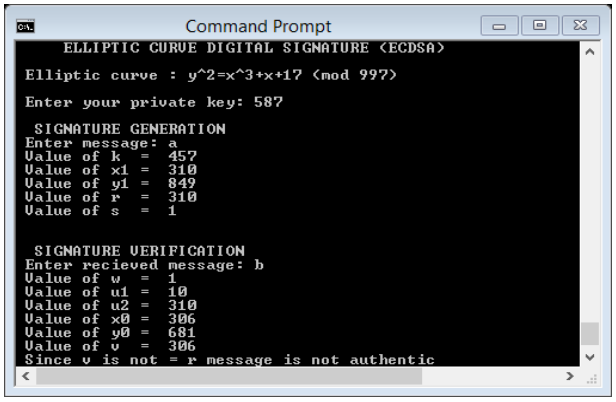


Figure 6: Signature not verified V≠R for Private key =587

To assure variation the algorithm was run multiple ti
To assure variation the algorithm was run multiple times on different sets of data. The mean of five readings for each input set is tabulated. For representation of workload characteristics, six different sets of inputs of lengths in the range 10^1 to 10^9 were taken. Table 3 lists the time required in seconds for hashing of inputs using SHA-1.
ECDSA is observed to be very fast and efficient algorithm. It can generate and verify the signature in seconds for a very large message. Table 2 and Table 3 shows the variation of signature generation and signature verification time and length of input respectively.

Table 2: Time needed for ECDSA signature generation

Size of input (n characters)	Time (t in seconds)
10^1	$8.042*10^{-3}$
10^3	$1.028*10^{-2}$
10^5	$2.354*10^{-2}$
10^7	$2.472*10^{-1}$
10^8	1.521
10^9	13.548

Table 3: Time needed for ECDSA signature verification

Size of input (n characters)	Time (t in seconds)
10^1	$6.002*10^{-3}$
10^3	$8.103*10^{-3}$
10^5	$1.151*10^{-2}$
10^7	$9.846*10^{-2}$
10^8	$8.829*10^{-1}$
10^9	8.924

It can also be seen that the time required to sign a message increases drastically as message size increases from 10^8 characters to 10^9 characters. The time needed to generate and verify the digital signature varies linearly with size of message. Also, the signature verification is faster process when compared to signature generation. The time required for ECDSA is in few seconds even for a very large message this makes the algorithm desirable for use in devices with constrained resources and limited memory size like smart cards.

VIII. CONCLUSION AND FUTURE SCOPE

In this research paper the secure digital signature algorithm for low memory devices is proposed. This method is based on Galois field used in ECDSA for the speed up the encryption and decryption methods used in digital signature. Digital signatures serve the purpose of validation and authentication of documents and messages. The of proposed Digital signature generation and verification algorithm using ECDSA is stated and implemented in programming language Python. The aim is to develop stronger and faster security mechanism for authentication and integrity of data. The key storage needed for ECDSA is minimal because even a smaller key length provides equivalent security when compared to DSA key. The small size of the key makes the process of key generation and key exchange faster. If a single character of the message is changed, the digital signature becomes redundant and it has a strong avalanche effect. The process of confusion and diffusion is also implemented and shown in the results. Time analysis of ECDSA shows that it is a fast, efficient and secure algorithm. These advantages make ECDSA ideal for use in WSN, PDAs, cellular phones and smart cards that having constrained environments, limited power and low memory.

REFERENCES

[1] H. Modares, M. T. Shahgoli, H. Keshavarz, A. Moravejosharieh, R. Salleh. "Make a Secure Connection Using Elliptic Curve Digital Signature". International Journal of Scientific & Engineering Research IJSER Volume 3, Issue 9, Pages 1-8, September 2012.
[2] B. B. Brumley, M. Barbosa, and F. Vercauteren. "Practical realisation and elimination of an ECC related software bug attack" In the proceeding of Cryptographers' Track at the RSA Conference CT-RSA, volume 7178 of LNCS Springer, Berlin Pages 171–186, 2012.

[3] D. Boneh, H. Shacham, "Group signatures with verifier-local revocation", 11th ACM Conference on Computer and Communications Security CCS, Washington DC USA, Pages168–177, 2004.
[4] D. J. Bernstein. Curve25519: New Diffie-Hellman speed records. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, Public Key Cryptography – PKC 2006, volume 3958 of LNCS, pages 207–228. Springer, 2006.
[5] I. Biehl, B. Meyer, and V. M¨uller. Differential fault attacks on elliptic curve cryptosystems. In M. Bellare, editor, CRYPTO, volume 1880 of Lecture Notes in Computer Science, pages 131–146. Springer, 2000.
[6] Hongjie Zhu, Daxing Li "Research on Digital Signature in Electronic Commerce" Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong
[7] Mohammad Noor Nabi, Sabbir Mahmud, And M Lutfar Rahman "Implementation and Performance Analysis Of Elliptic Curve Digital Signature Algorithm", Daffodil International University journal of science and technology, volume 2, issue 1, Page 28-33 January 2007.
[8] Aqeel Khalique, Kuldip Singh, Sandeep Sood "Implementation Of Elliptic Curve Digital Signature Algorithm" International Journal Of Computer Applications, Volume 2, Issue 5, Page 1263-1271, May 2010.
[9] D. Jetchev and R. Venkatesan. Bits security of the elliptic curve Diffie-Hellman secret keys. In D. Wagner, editor, CRYPTO, volume 5157 of LNCS, pages 75–92. Springer, 2008.
[10] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov. The most dangerous code in the world: Validating SSL certificates in non-browser software. In T. Yu, G. Danezis, and V. D. Gligor, editors, ACM Conference on Computer and Communications Security, pages 38–49. ACM, 2012.
[11] Joppe W. Bos, Craig Costello, Patrick Longa, Michael Naehrig, Selecting elliptic curves for cryptography: An efficiency and security analysis, Journal of cryptographic Engineering, November 2016, Volume 6, issue 4, page 259-286, Springer 2016.
[12] Paramjit Kaur, RakeshKumar and Harinder Kaur, "*An Improved Security Network Life Based on Data Ant Colony Optimization Method Used in Wireless Mesh Network*", International Journal of Computer Sciences and Engineering, Vol.6, Issue.3, pp.50-56, 2018.
[13] Abhishek Satyarthi, Sanjiv Sharma, "*A Review of Homomorphic Encryption Algorithm for Achieving Security in Cloud: Review Article*", International Journal of Computer Sciences and Engineering, Vol.5, Issue.6, pp.19-23, 2017.
[14] Kodge B. G., "*Information Security: A Review on Steganography with Cryptography for Secured Data Transaction*", International Journal of Scientific Research in Network Security and Communication, Vol.5, Issue.6, pp.1-4, 2017
[15] Book References William Stallings, "Cryptography and Network Security", Prentice Hall, 5th Edition, Pages 285-296, 2010.
[16] Kaufman, c., Perlman, R., and Speciner, M., "Network Security, Private Communication in a public world", 2nd Edition. Prentice Hall Print, Pages 274-282, 2002.
[17] Behrouz A Forouzan, "Cryptography and Network Security", McGraw Hill, 2nd Edition, Pages 290-297, 2010.
[18] Bruce Schneier, "Applied Cryptography: Protocols, Algorithms and source code in C", Wiley publishing Inc. 2nd Edition, Pages 201-235, 2015.

Author's Profile

Dr. Mohd. Amjad is currently working as Associate Professor in the Department of Computer Engineering, F/o Engineering & Technology, Jamia Millia Islamia (Central University), New Delhi. He received B.Tech. Degree from A.M.U. Aligarh in computer Engineering, M.Tech. Degree in Information Technology from GGSIP University New Delhi and Ph.D. from Jamia Millia Islamia, from the Department of Computer Engineering New Delhi. He has published more than 50 research papers in reputed international journals including Thomson Reuters (SCI & Scopus) and conferences including IEEE and it's also available online. His research interests includes Network Security, Internet and mobile computing, Mobile Ad hoc Networks and wireless sensor networks. Dr. Amjad has more than 16 Years of teaching experience at U.G and P.G. Level.

