
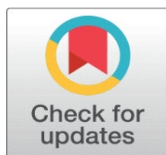


# MATHEMATICAL MODEL OF DIGITAL SIGNATURE BASED ON ECDSA AND SCHMIDT-SAMOA CRYPTOSYSTEM

Bhavadip Moghariya <sup>1</sup>, Ravi Gor <sup>2</sup>

<sup>1</sup> Research Scholar, Department of Applied Mathematical Science, Actuarial Science and Analytics, Gujarat University, India

<sup>2</sup> Department of Applied Mathematical Science, Actuarial Science and Analytics, Gujarat University, India



**Received** 01 January 2024

**Accepted** 16 February 2024

**Published** 01 March 2024

## Corresponding Author

Bhavadip Moghariya,  
[bhavadipmoghariya@gujaratuniversity.ac.in](mailto:bhavadipmoghariya@gujaratuniversity.ac.in)

**DOI** [10.29121/IJOEST.v8.i1.2024.572](https://doi.org/10.29121/IJOEST.v8.i1.2024.572)

**Funding:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Copyright:** © 2024 The Author(s). This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

With the license CC-BY, authors retain the copyright, allowing anyone to download, reuse, re-print, modify, distribute, and/or copy their contribution. The work must be properly attributed to its author.



## ABSTRACT

Digital Signature Technology is replacing paper-based work for customers and employees in various industries and e-commercial environment. Digital Signature provides cryptographic services like authentication, non-repudiation, and integrity for the digital data. With the development of internet, Digital Signature becomes increasingly important for security because of its integrity and authenticity. It is an electronic signature that can be used to authenticate the identity of the sender. Digital Signature does not provide confidentiality until an encryption algorithm is applied. In this study, a new model of Digital Signature is introduced using the Elliptic Curve Digital Signature Algorithm (ECDSA) with an encryption technique Schmidt Samoa Cryptosystem. This model provides double layer security with encryption as well as signing protocol. Proposed model provides features like confidentiality, non-repudiation, and authenticity.

**Keywords:** Digital Signature, ECDSA, Schmidt-Samoa Cryptosystem (SSC)

## 1. INTRODUCTION

In E-commerce business and cloud technology, authentication, and confidentiality are very important property. Similarly, non-repudiation is the important property of information security in blockchain technology. Digital Signature is cryptographic tool to verify the integrity of a file or a message. Different methods and algorithms of cryptography have been used to achieve these types of goals.

The purpose of Digital Signature is the same as handwritten signature. Instead of using pen and paper, a Digital Signature can be generated using digital keys. An essential benefit of Digital Signature is that one cannot generate fake Digital Signature. Digital signature is always a part of shared document or data.

Some of the standard approach was defined by National Institute of Standards and Technology (NIST) of US called Digital Signature Standard (DSS). Digital Signature Algorithm (DSA) is the part of DSS developed by NIST. Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the Digital Signature Algorithm (DSA).

Elliptic Curve Digital Signature Algorithm (ECDSA) is a public key cryptosystem which is based on Elliptic Curve Cryptography (ECC). In ECC, Elliptic curves are defined over finite field and algebraic structure of these elliptic curves are used for Digital signature algorithms. Discrete logarithm problem (DLP) is defined on an elliptic curve. Without knowing some parameters, it is very unlikely to solve DLP which gives strong security for an algorithm based on ECC.

Compared to other techniques, ECDSA has a relatively high computational complexity, which provides superior protection against various kinds of attacks. So, ECDSA is frequently used in wide range of applications. ECDSA is based on Elliptic Curve Cryptography, which requires some different algebraic operations.

**Mathematical Model of ECDSA**

An Elliptic Curve  $E$  is defined over a field  $K = F_p$ . Which is the set of points satisfying an equation  $y^2 = x^3 + ax + b$ , where  $a, b \in K$  and  $4a^3 + 27b^2 \neq 0 \pmod p$ .

Here, characteristic of  $K = F_p$  is neither 2 nor 3.

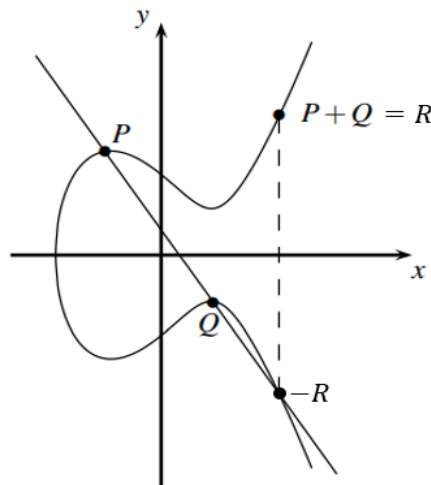
Different values of  $a$  and  $b$  gives different elliptic curves. These elliptic curves also contain a special point  $O$ , called the point at infinity.

Set of points on Elliptic Curve form a group under a specific binary operation. This binary operation is defined over finite fields. The main operation, point multiplication is achieved by two basic elliptic curve operations.

- (1) Point addition
- (2) Point doubling

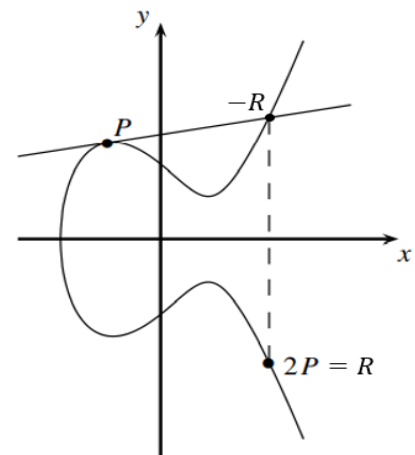
**1) Point addition**

**Figure 1**



**Figure 1 (Point Addition)**

**Figure 2**



**Figure 2 (Point Doubling)**

Consider two distinct points  $P$  and  $Q$  on an elliptic curve as shown in [Figure 1](#). If  $Q \neq -P$  then a line drawn through the points  $P$  and  $Q$  will intersect the elliptic curve at exactly one more point  $(-R)$  (negative of  $R$ ). The reflection of the point  $(-R)$  with respect to  $X$  - axis gives the point  $R$ , which is the addition of points  $P$  and  $Q$ . Thus, on an elliptic curve,  $P + Q = R$ . If  $Q = -P$ , the line through this point does not intersect to Elliptic Curve at any point. So, it is considered that line intersects a point at infinity  $O$ . Hence,  $P + (-P) = O$ . Negative of a point is the reflection of that point with respect to  $X$  - axis.

**(2) Point doubling**

Point doubling is the addition of a point  $P$  on the elliptic curve to itself. It obtains another point  $R$  on the same elliptic curve. i.e.,  $R = 2P$ .

Consider a point  $P$  on an elliptic curve as shown in [Figure 2](#). If  $y$  coordinate of the point  $P$  is nonzero, then the tangent line at  $P$  will intersect the elliptic curve at exactly one more point say  $(-R)$ . The reflection of the point  $(-R)$  with respect to  $X$  - axis gives the point  $R$ , which is the result of doubling the point  $P$ . i.e.,  $R = 2P$ .

If  $y$  coordinate of the point  $P$  is zero, then the tangent at this point intersects a point at infinity  $O$ . Hence,  $2P = O$  when  $y_j = 0$ .

Algebraically, addition can be defined as follows:

Let  $P(x_1, y_1)$  and  $Q(x_2, y_2)$  are two points on the elliptic curve then,

$$R = P + Q = \begin{cases} O & \text{if } x_1 = x_2 \\ Q & \text{if } P = O \\ (x_3, y_3) & \text{otherwise} \end{cases}$$

$$\text{Where } x_3 = \begin{cases} \lambda^2 - x_1 - x_2, & \text{if } P \neq Q \text{ (Point addition)} \\ \lambda^2 - x_1, & \text{if } P = Q \text{ (Point doubling)} \end{cases}$$

$$\text{and } y_3 = \lambda(x_1 - x_3) - y_1$$

$$\text{Where } \lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q \text{ (Point addition)} \\ \frac{3x^2 + a}{2y_1}, & \text{if } P = Q \text{ (Point addition)} \end{cases}$$

Let an entity A wants to send a message to B then the steps are as follows:

Before the key generation, signer choose some parameters, prime numbers  $p, a$  and  $b$ . These parameters are used to define a specific elliptic curve.

Select a point  $P$  on curve  $y^2 = x^3 + ax + b$  which generates a group of prime order  $n$ . Sometimes point also known as base point.

**1) Key Generation**

- Select a random integer  $d$  in the interval  $[1, n - 1]$
- Compute  $Q = dP$
- A's public key is  $Q$  and A's private key is  $d$

**2) Signature Generation**

- Select a random integer  $k$  in the interval  $[1, n - 1]$
- Compute  $kP = (x_1, y_1)$  and  $r = x_1 \pmod n$  (where  $x_1$  is regarded as an integer between 0 and  $q - 1$ ). If  $r = 0$ , then reselect  $k$
- Compute  $t = k^{-1}$

- Compute,  $s = k^{-1}(H(m) + dr) \pmod{n}$  where  $H(m)$  is the hash value compute by secure hash algorithm. If  $s = 0$ , then reselect  $k$
- The signature for the message  $m$  is the pair of integers  $(r, s)$

### 3) Signature Verification

- Obtain an authenticated copy of sender's public key  $Q$
- Verify that the integers  $r$  and  $s$  are in the interval  $[1, n - 1]$
- Compute  $w = s^{-1} \pmod{n}$  and  $H(m)$
- Compute  $u_1 = H(m)w \pmod{n}$  and  $u_2 = rw \pmod{n}$
- Compute  $u_1P + u_2Q = (x_0, y_0)$  and  $v = x_0 \pmod{n}$
- Accept the signature if and only if  $v = r$

### SSC encryption technique

SSC algorithm introduced by [Schmidt-Samoa \(2005\)](#). It is based on difficulty of integer factorization as prime numbers.

SSC encryption technique also has three steps:

(1) Key Generation                      (2) Encryption                      (3) Decryption

#### 1) Key Generation

- Choose two distinct large prime numbers  $p$  and  $q$ .
- Calculate  $N = p^2q$ .
- Calculate  $d = N^{-1} \pmod{\text{lcm}(p-1, q-1)}$ .
- Public key is  $N$  and private key is  $d$ .

#### 2) Encryption

- Compute  $C = m^N \pmod{N}$
- Cipher text is  $C$  and send it to the receiver.

#### 3) Decryption

- Compute  $m = C^d \pmod{pq}$
- $m$  is the required plaintext.

## 2. LITERATURE REVIEW

[Thangavel & Varalakshmi \(2016\)](#) proposed a novel and efficient public key cryptosystem, enhanced Schmidt Samoa (ESS) to safeguard the data confidentiality in the cloud. The ESS cryptosystem deal with composition of four prime numbers, which increase the complexity to break the cryptosystem compared to SSC. The performance of the ESS cryptosystem has been tested with brute force attack and integer factorisation method. Performance analysis highlights that the execution time for ESS encryption as well as ESS decryption comparatively too lesser than SSC encryption and decryption in in both cases (i) file size variation and (ii) key size variation.

[Al-Haija et al. \(2018\)](#) provide a systematic review of Schmidt Samoa Cryptosystem to help crypto designers in efficient implementation in hardware or software-based applications.

[Panjwani & Mehta \(2015\)](#) presented the hardware-software co-design of ECDSA, where the focus has been to reduce the hardware resource utilisation and increase the throughput of the signature generation and verification process. The main module which has been implemented in software is the key generation of

ECDSA. Proposed implementation is comparatively faster than other implementations. Significant timing advantage has been achieved due to wordwise Montgomery multiplication implementation over both  $GF(2^{163})$  and  $GF(n)$  which are basic building blocks of the entire implementation.

[Khalique et al. \(2010\)](#) describes the implementation of ECDSA over elliptic curve P-192 and discusses related security issues. ECDSA has no sub exponential algorithm to solve the elliptic curve discrete logarithm problem on a properly chosen elliptic curve. So, it takes full exponential time. The key generated in the implementation is highly secured and it consumes lesser bandwidth because of small key size used by the elliptic curves.

[Farooq et al. \(2019\)](#) proposed a light-weight Elliptic Curve Digital Signature Algorithm. Which is certificate-based authentication mechanism by utilizing different elliptic curves. It has been shown that ECDSA authentication scheme has much better timing performance compared to RSA and can be safely used for authentication in Advanced Metering Infrastructure.

[Koblitz et al. \(2000\)](#) introduced Elliptic Curve Cryptography. Further, Elliptic Curve Cryptosystems are defined based on the discrete logarithm problem. This paper discussed Elliptic Curves and different operations of points over finite field.

[Zhang et al. \(2011\)](#) proposed the improved digital signature algorithm based on the elliptic curve cryptography and enhance the security of the digital signature. This proposed method increased one step that encrypt signature with signer's private key and then sent the encrypted result to the verifier. Verifier verifies the encrypted result before verifying the signature.

[Kavin & Ganapathy \(2021\)](#) proposed an Enhanced Digital Signature Algorithm (EDSA) for verifying the data integrity while storing the data in cloud database. Proposed EDSA had been developed according to the Elliptic Curve Square points that were generated by using an upgraded equation and these points were used as public key. Also, a new base formula was introduced for signing and verification process. This work introduced a new compression technique which had been used for reducing the bit size of the signature.

### 3. PROPOSED METHOD

A sender wants to send a message to a receiver. Then the model work as given below:

**Step 1:** Sender chooses parameters of ECDSA algorithm and receiver chooses parameters of SSC technique.

**Step 2:** Using above parameters sender generates a key for signing and receiver generates a key for encryption and share this key to sender.

**Step 3:** Using the key shared by receiver, sender encrypts the message by SSC technique.

**Step 4:** Sender generates signature for this encrypted message by ECDSA.

**Step 5:** Sender sends this signature to receiver along with encrypted message and key which are required for signature verification.

**Step 6:** Receiver verifies the signature by ECDSA.

**Step 7:** If signature is verified, receiver decrypts the cipher text by SSC technique to read the original message. If it is not verified, then someone has forged the original data.

#### 4. NUMERICAL EXAMPLE OF THE PROPOSED METHOD

Let sender Bob wants to send a message 'm = 120' to receiver Alice. Then Bob and Alice will follow the following phases and steps.

- **Phase I: Selection of Parameters**

**Step 1:** Bob chooses following parameters for ECDSA: prime  $p = 43$ ,  $a = 3$ ,  $b = 11$ . So, elliptic curve becomes  $y^2 = x^3 + 3x + 11$ . Select generator point  $P = (8,17)$ . Then order of group generated by  $P$  is  $n = 13$  which is prime.

- **Phase II: Key Generation**

**Step 1:** Bob chooses  $d = 7$ , where  $1 \leq d \leq 12$  and calculate  $Q = dP = 7(8,17) = (4,42)$ .

So, private key  $d = 7$  and public key  $Q = (4,42)$  for ECDSA.

**Step 2:** Alice chooses prime  $\alpha = 71$  and  $\beta = 89$ .

Calculate  $N = \alpha^2\beta = (71)^2 * 89 = 448649$ .

$N = 448649$  is the public key for SSC encryption technique.

Alice share this public key to Bob for encryption.

**Step 3:** Find  $lcm(\alpha - 1, \beta - 1) = lcm(70,88) = 3080$ .

Calculate  $d_e = N^{-1}(\text{mod } 3080) = (448649)^{-1}(\text{mod } 3080) = 2369$ .

$d_e = 2369$  is the private key for SSC encryption technique.

- **Phase III: Signature Generation**

**Step 1:** For ECDSA, Bob chooses  $k_E = 9$ , where  $1 \leq k_E \leq 12$ .

Calculate  $k_E P = (x_1, y_1) = 9(8,17) = (30,21)$ . So,  $r = x_1 = 30 = 4 (\text{mod } 13)$ .

**Step 2:** To encrypt the message by SSC encryption technique,

Bob encrypt the message  $m = 120$ , by calculating  $e = m^N (\text{mod } N)$

$e = 120^{448649} (\text{mod } 448649) \therefore e = 81094$

So, encrypted message is  $e = 81094$ .

**Step 3:** Compute the Hash value of encrypted message,

$H(e) = H(81094) = (5d24767acba10f2762887ca5a1620bec)_{16}$

$H(e) = (123807529348093653224751694692915219436)_{10}$ .

Compute  $s = k_E^{-1}(H(e) + d * r) (\text{mod } n)$

$= 3 * (123807529348093653224751694692915219436 + 7 * 4) (\text{mod } 13)$

$\therefore s = 3 (\text{mod } 13)$

So, signature is  $(r, s) = (4,3)$ .

Bob shares encrypted message  $e = 81094$  and signature  $(r, s) = (4,3)$  along with public key  $Q = (4,42)$ ,  $P = (8,17)$  and parameters  $a = 3$ ,  $b = 11$ ,  $p = 43$ ,  $n = 13$ .

- **Phase IV: Signature Verification and Decryption of Cipher text**

- **Signature Verification**

**Step 1:** Alice verifies that  $r = 4$  and  $s = 3$  lies in the interval  $[1,12]$  ( $= [1, n - 1]$ ).

Find the hash value of encrypted message  $e = 81094$ .

which is,

$$H(81094) = (5d24767acba10f2762887ca5a1620bec)_{16}$$

$$= (123807529348093653224751694692915219436)_{10}$$

**Step 2:** Now, calculate  $w = s^{-1} = 3^{-1} = 9 \pmod{13}$ .

**Step 3:** Calculate  $u_1 = H(e) * w$

$$u_1 = 123807529348093653224751694692915219436 * 9 \pmod{13} \therefore u_1 = 4 \pmod{13}$$

Also,  $u_2 = r * w = 4 * 9 \pmod{13} = 10 \pmod{13}$ .

**Step 4:** Find  $(x, y) = u_1P + u_2Q = 4(8,17) + 10(4,42) = (30,21)$ .

So,  $x = 30 = 4 \pmod{13}$ .

**Step 5:** Verify that  $r = x$  or not. Here,  $r = 4 = x$ .

So, signature is verified.

If signature is not verified, someone has forged the shared data.

- **Decryption of Message**

**Step 6:** If Signature is verified, decrypt the cipher text by SSC encryption technique to read the original text.

$$\text{Compute } m = C^{d_e} \pmod{\alpha\beta} = (81094)^{2369} \pmod{6319} = 120.$$

So, original message is  $m = 120$ .

Hence, using this method Alice got the unforged and signed message sent by Bob.

## 5. TIME COMPLEXITY IN RUN TIME OF PROPOSED SCHEME

Table 1 and Table 2 shows the run time of Digital Signature scheme using ECDSA and SSC encryption technique. Here, encryption and decryption time for SSC encryption techniques, signature generation and verification time for ECDSA, signature generation and verification time for proposed signature scheme and combined time for encryption and signature generation as well as signature verification and decryption time is described. These timings are denoted for different lengths of messages. Here, we considered standard elliptic curve NIST256 and SHA-256.

**Table 1**

Table 1			
Algorithm	Process	Message Length (in Bit)	
		128	256
SSC (ms)	Encryption	3006.11457824707	6207.09295272827
	Decryption	4.26497459411621	8.42375755310058
ECDSA (ms)	Signing	1.1199951171875	1.08008384704589
	Verification	3.89752388000488	4.38675880432128
Proposed Scheme (ms)	Signing	3007.4019908905	6208.39200019836
	Verification	8.02445411682129	12.6346588134765
Combined (ms)	Signing + Encryption	3007.23457336425	6208.17303657531
	Verification + Decryption	8.16249847412109	12.8105163574218



**Table 2**

Table 2			
Algorithm	Process	Message Length (in Bit)	
		512	1024
SSC (ms)	Encryption	12854.2956829071	26501.6244411468
	Decryption	17.8483009338378	36.2663269042968
ECDSA (ms)	Signing	1.08251571655273	1.07722282409668
	Verification	4.04844284057617	3.97439002990722
Proposed Scheme (ms)	Signing	12855.621767044	26502.9312133788
	Verification	21.9624042510986	40.7961368560791
Combined (ms)	Signing + Encryption	12855.3781986236	26502.7016639709
	Verification + Decryption	21.896743774414	40.2407169342041

From the tables, a vast difference between SSC encryption time and decryption time has been observed. Also, encryption and decryption time is highly influenced by the size of message.

From Chart 1, it is observed that there is a significant difference between SSC encryption time and ECDSA signature generation time. Also, encryption time and signature generation time for proposed method is significantly increasing as size of message is increased. A positive result is that the addition of SSC encryption time and ECDSA signature generation time is almost equal to the signature generation time of the proposed scheme.

**Chart 1**



**Chart 1**

From Chart 2, it is observed that signature verification time is increased with size of messages increased. The difference between verification time of proposed method and combined time of ECDSA verification and SSC decryption is almost zero.



Chart 2

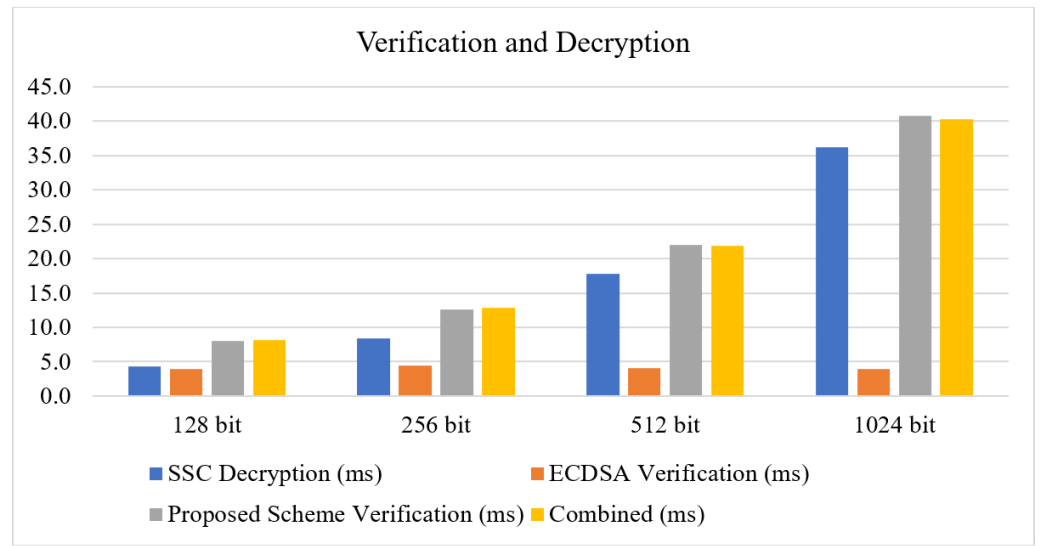


Chart 2

## 6. CONCLUSION

In this paper, proposed method uses Elliptic Curve Digital Signature Algorithm (ECDSA) to generate signature and Schmidt-Samoa Cryptosystem (SSC) to encrypt the message. This method preserve data confidential because of encryption algorithm. It prevents the forgery of shared data and the receiver can also detect any forgery in shared data if it happens. Additionally, this approach offers non-repudiation and authenticity because of ECDSA. Proposed scheme provide strong security against various attacks as there is no plaintext in calculation of algorithm as well as in the sharing.

The Digital Signature scheme based on ECDSA and SSC encryption technique is very effective for all sizes of messages. The signature generation time by proposed method is almost same to the addition of SSC encryption time and ECDSA signature generation time. The same case happens for signature verification and decryption.

## CONFLICT OF INTERESTS

None.

## ACKNOWLEDGMENTS

None.

## REFERENCES

- Al-Haija, Q. A., Asad, M. M., & Marouf, I. (2018). "A Systematic Expository Review of Schmidt-Samoa cryptosystem". *International Journal of Mathematical Sciences and Computing (IJMSC)*, 4(2), 12-21. <https://doi.org/10.5815/ijmsc.2018.02.02>
- Farooq, S. M., Hussain, S. S., & Ustun, T. S. (2019, March). "Elliptic Curve Digital Signature Algorithm (ECDSA) Certificate-Based Authentication Scheme for Advanced Metering Infrastructure". In *2019 Innovations in Power and*

- Advanced Computing Technologies (i-PACT), 1, 1-6. IEEE. <https://doi.org/10.1109/i-PACT44901.2019.8959967>
- Hieu, M. N., & Tuan, H. D. (2012, October). "New Multisignature Schemes with Distinguished Signing Authorities". In The 2012 International Conference on Advanced Technologies for Communications, 283-288. <https://doi.org/10.1109/ATC.2012.6404277>
- Jarusombat, S., & Kittitornkun, S. (2006). "Digital Signature on Mobile Devices Based on Location." In 2006 International Symposium on Communications and Information Technologies, IEEE, 866-870. <https://doi.org/10.1109/ISCIT.2006.339860>
- Kavin, B. P., & Ganapathy, S. (2021). "A New Digital Signature Algorithm for Ensuring the Data Integrity in Cloud Using Elliptic Curves". The International Arab Journal of Information Technology, 18(2), 180-190. <https://doi.org/10.34028/iajit/18/2/6>
- Khalique, A., Singh, K., & Sood, S. (2010). "Implementation of Elliptic Curve Digital Signature Algorithm". International Journal of Computer Applications, 2(2), 21-27. <https://doi.org/10.5120/631-876>
- Koblitz, N., Menezes, A., & Vanstone, S. (2000). "The State of Elliptic Curve Cryptography", Designs, Codes and Cryptography, 19, 173-193. <https://doi.org/10.1023/A:1008354106356>
- Neal, K. (1985). "Elliptic Curve Cryptosystems", Mathematics of Computation, 48(177), 203-209. <https://doi.org/10.1090/S0025-5718-1987-0866109-5>
- Paar, C., & Pelzl, J. (2009). "Understanding Cryptography: A Textbook for Students and Practitioners". Springer Science & Business Media. <https://doi.org/10.1007/978-3-642-04101-3>
- Panjwani, B., & Mehta, D. C. (2015, August). "Hardware-Software Co-Design of Elliptic Curve Digital Signature Algorithm Over Binary Fields". In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 1101-1106. <https://doi.org/10.1109/ICACCI.2015.7275757>
- Rahat, A., & Mehrotra, S. C. (2011). "A Review on Elliptic Curve Cryptography for Embedded Systems". International Journal of Computer Science & Information Technology (IJCSIT), 3(3). <https://doi.org/10.5121/ijcsit.2011.3307>
- Schmid, M. (2015). "ECDSA-Application and Implementation Failures".
- Schmidt-Samoa, K. (2005). "A New Rabin-Type Trapdoor Permutation Equivalent to Factoring and Its Applications". Cryptology ePrint Archive. <https://doi.org/10.1016/j.entcs.2005.09.039>
- Sowmiya, B., Poovammal, E., Ramana, K., Singh, S., & Yoon, B. (2021). "Linear Elliptical Curve Digital Signature (LECDs) with Blockchain Approach for Enhanced Security on Cloud Server". IEEE Access, 9, 138245-138253. <https://doi.org/10.1109/ACCESS.2021.3115238>
- Stinson, D. R. (2005). "Cryptography: Theory and Practice". Chapman and Hall Book, CRC Press. <https://doi.org/10.1201/9781420057133>
- Thangavel, M., & Varalakshmi, P. (2016). "Enhanced Schmidt-Samoa Cryptosystem for Data Confidentiality in Cloud Computing". International Journal of Information Systems and Change Management, 8(2), 160-188. <https://doi.org/10.1504/IJISCM.2016.079567>
- Timothy, D. P., & Santra, A. K. (2017, August). "A Hybrid Cryptography Algorithm for Cloud Computing Security". International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS), 1-5. <https://doi.org/10.1109/ICMDCS.2017.8211728>

- Utama, K. D. B., Al-Ghazali, Q. R., Mahendra, L. I. B., & Shidik, G. F. (2017, October). "Digital Signature Using MAC Address-Based AES-128 and SHA-2 256-bit". International Seminar on Application for Technology of Information and Communication (iSemantic), 72-78. <https://doi.org/10.1109/ISEMANTIC.2017.8251846>
- Zhang, Q., Li, Z., & Song, C. (2011, August). "The Improvement of Digital Signature Algorithm Based on Elliptic Curve Cryptography". In 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), IEEE, 1689-1691.