INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

A Chosen-Provably-Secure Attack-Resistant Light-Weight Digital Signature Based on Elliptical Curve for Resource Constrained Applications

Dhanashree K Toradmalle[1]*, K Amarendra[2]

1 Associate Professor, Department of Computer Engineering, K J Somiaya Institute of Technology, Sion, Mumbai-77, Maharashtra, India
2 Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, 522502, Andhra Pradesh, India

OPEN ACCESS

Received: 23-02-2023
Accepted: 30-03-2023
Published: 21-04-2023

Citation: Toradmalle DK, Amarendra K (2023) A Chosen-Provably-Secure Attack-Resistant Light-Weight Digital Signature Based on Elliptical Curve for Resource Constrained Applications. Indian Journal of Science and Technology 16(16): 1205-1213. https://doi.org/10.17485/IJST/v16i16.410

* **Corresponding author**.

dhanashree.t@somaiya.edu

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To build a bridge to provide a solution by developing a lightweight ECDSA method that is not only lower in computational aspect but also is more secure than the Zhong's ECDSA. **Methods:** The proposed research work performs cryptanalysis of Zhong's ECDSA and demonstrates that the method fails to withstand MITM attacks. The proposed ECDSA uses only 1 elliptic curve point multiplication ECPM operation, 1 Modular multiplication operation and zero Modular Inverse operations making it lightweight in terms of computational time. Zero inverse operations save computational time as the process involves scalar mathematics which is time-consuming. **Findings:** Zhong's ECDSA is not secure. Additionally, the technique takes 13.28% less time to sign data than the suggested ECDSA method. Through proofs, it is shown by comparison of the proposed ECDSA and Zhong's ECDSA and cryptanalysis that the proposed ECDSA is more applicable in real time. Although Zhong's Method for Signature verification at the Receiver end takes 8.2% more time, the recommended technique stands out in comparison to Zhong's ECDSA w.r.t security. **Novelty:** The work is a detailed expression of the provably-secure attack-resistant light-weight digital signature based on elliptical curve for resource constrained applications. Advancing, the novelty of the work lies in the comparison of the two techniques w.r.t their performance parameters like number of keys generated, time taken to generate keys, number of keys verified, time taken for key verification, time taken for Signature generation and time taken for Signature verification.

Keywords: Digital signature; MITM; ECDSA; Replay attacks; Forgery attacks

1 Introduction

Today, strong cryptographic operations that are a component of cryptosystems are crucial to information security[1,2], big data security[3] and wireless networks.

Advanced study[4] requires the ECDSA toughening with the incorporation of advance secrecy where even if the Signer's enigmatic key is found today, digital signatures can secure previously marked messages. The toughened ECDSA should be able to withstand security threats like man-in-the-middle attacks, replay attacks and forgery attacks. The ECDSA algorithm is appropriately used in WSN[5], RFID[6], smart card[7], adhoc networks[8] and IoT[9–13] implementations due to its performance and security. ECDSA digital signatures are more effective than DSA and RSA one's in constrained-resource devices as they work on lower key sizes. Numerous writers have suggested utilizing ECDSA in resource-constrained situations (memory, energy, and CPU capability). To avoid manipulation with ECDSA signatures, security measures are proposed. Many techniques are devised to withstand attacks and thereby improve the security issues in networks[14]. However, in terms of time, storage, and sophisticated computations, these defenses are too expensive. By developing novel digital signature schemes based on elliptic curves, security targets can be met, such as confidentiality, legitimacy, and non-revocation. When creating an ECDSA scheme, the computational time is the main worry, and it also depends on the number of ECC and inverse operations. Researchers now face a challenge as they work to update ECDSA while maintaining security and computation time as their top priorities. Thus, maintaining the need of a low, computationally inexpensive ECDSA based scheme which is also secured is the objective of each researcher today. This has been a major topic of concern in spite of much work done in the direction of ECDSA enhancements.

Hong Zhong's[15] ECDSA is a work in the same direction to reduce the computational expense required during the process of creating and verifying signatures, by making an effort to achieve strength by omitting the inverse standard operations. The challenge of computational time and speed are considered by Zhong's scheme but the void is security for which the research article proposes a novel technique which is further validated for attacks. The research article presents the Zhong's ECDSA method, performs cryptanalysis over Zhong's method and also evaluates its performance. Further the work also proposes an Improved ECDSA method which is lightweight in terms of the ECC operations and zero inverse computations. The improved ECDSA is proved to be more secure than its predecessor. The performance parameters of Zhong's ECDSA and proposed ECDSA are compared using NIST ECDSA standards.

2 Methodology

2.1 Zhong's ECDSA Scheme

The middleman or intrusive party can rapidly change or replace the message that the recipient cannot understand by changing the hash value. Zhong's scheme[15] aims to improve efficiency by reducing the reserve standard inverse operations, but it is insecure because it does not satisfy the security requirements for a digital signature scheme since it is vulnerable to a hacker completely changing the message and replacing the current message hash value with a different hash value. The notations used are as follows:

G: Basepoint of elliptic curve
d: Private key of Alice
m: message
e: hash value of message m

2.1.1 ***Signature Generation Phase***

When Alice sends the message to Bob, and so obtains a digital signature r, s which is generated by the following steps:

Step 1: Select a random k in the range of [1, n - 1]
Step 2: Compute a curve point k * G = (x1, y1)
Step 3: Compute value of r = x1 mod n. If r = 0, then go back to step 1
Step 4: Compute the value of e = SHA -1(m)
Step 5: Compute the value of s = (e +k + r d) mod n. If s = 0, then return to step1
Step 6: Send the message m and computed digital signature (r, s)

2.1.2 ***Signature Verification Phase***

Following these steps, Bob validates the digital signature:

Step 1: Confirm that r and s are integers in [1, n-1]. If not, the signature is Invalid
Step 2: Calculate e = SHA -1(m)
Step 3: Calculate w = (s - e) mod n
Step 4: Ascertain a curve X= w * G – r * Q = (x1, y1)
Step 5: On the off chance that If X=0, the digital signature is invalid else ascertain
v = x1 mod n.

Step 6: Bob will acknowledge the digital signature if and only if v = r

2.2 Cryptanalysis of Zhong's ECDSA scheme

By simply adding the hash value, the Middle Man or intruder can easily change or supersede the message that the receiver cannot interpret. Let m1 be the message of the middle man, which is modified or replaced by the original message m, whose hash values e1 and e respectively. The following is a full discussion of the cryptanalysis of Zhong's scheme, which demonstrates how Zhong's strategy favors man-in-the-middle attacks.

The following is an account of the attack:

1. Compute hash value e of the message m
2. Compute signature for message m, s = e + k + r d
3. New/modified message m1
4. Compute hash value e1 of the message m1
5. Compute signature for new message m1, s1= s - e + e1
6. (s1, x1) is the signature for the message m1.
7. Substitute the value of s from step 2 in step 5 we get,

s1 = e + k + d – e + e1

where s1 is Middle Man's signature element.

Hence, a hacker can change the message's hash value and add new data without knowing the Sender's or the Receiver's private or public keys. Security is at risk because the receiver cannot recognize this alteration. One of the most significant weaknesses in the Man in the Middle assault is revealed as the security of Hong Zhong's strategy is investigated. The system aims to increase effectiveness by decreasing reserve standard inverse operations, however it falls short of security due to the possibility of message modification and failure to meet the security requirements of a digital signature scheme.

2.3 Proposed Certificateless, Provably- Secure ECDSA

2.3.1 ***Stage of Key Generation***

Using generating point G and random integer number r the public key K is computed as follows:

1. Choose a random integer number r in the interval [0, n-1].
2. Compute K = r * G
3. The key-pair combination is (r, K) where r is the Private Key and K is the public key.

2.3.2 ***Stage of Signature Generation***

The Signer makes the following advances to sign message m using the domain parameter and private key:

1. Using$1 \leq p \leq n-1$ Select a random integer p (secret key)
2. The value of z = H(m) is ascertained
3. f = ((z + p) $\oplus$ (p + r)), where $\oplus$ is Ex-OR operation is ascertained
4. d = x-coordinator (f * G) is ascertained
5. Determine s = (z * r) + f mod n. If s = 0 then return to step 1
6. Signature for the message m is (d, s)

2.3.3 ***Stage of Signature Verification***

At the Receiver side, the message m ought to be validated with the following steps:

1. Firstly, confirm that s is an integer in the range [1, n −1]
2. Compute the hash value z of the message/document m
3. W = (x1, y1) = s * G – z * K
4. v = x-coordinate(W), finally, authenticate the signature by checking whether the equivalence v = d holds.

2.4 Security Proof of Proposed Certificateless, Provably- Secure ECDSA

2.4.1 ***MITM Attack***

If the signature for the message m is (d, s) and was generated by the authorized Sender, then s = (z * r) + f mod n is true. The following proof can be used to determine whether the algorithm is correct:

$$\begin{aligned} W &= s * G - z * K = ((z * r) + f) * G - z * K \\ &= z * r * G + f * G - z * K \\ &= z * K + f * G - z * K \\ &= f * G\,x - coordinate\,(W) \\ &= x - coordinate\,(f * G) \end{aligned} \tag{1}$$

As a result, v = d as a reason, the suggested technique by Hong Zhong et al, lacks to prevent the Man in the Middle attack

Sender: Bob Signature Generation

$$s = [\,(z * r) + f\ mod\ n] \tag{2}$$

Receiver: Alice Signature Verification

$$W = (x1, y1) = s * G - z * K \tag{3}$$

Intruder: Darth MITM Attack

$$s1 = [\,(z * r) + f\ mod\ n] - z + z1 \tag{4}$$

Darth tries to modify s1 from s but fails to achieve s1. Thus, Signature s1 fails on verification at Receiver Alice's end

2.4.1.1 **Signature verification**. At the Receiver side the message m ought to be validated with the following steps:

1. Firstly, confirm that s is an integer in the interim [1, n −1]
2. Compute the hash value z of the message/document m
3. W = (x1, y1) = s * G – z * K

W= {[(z * r) + f mod n] -z +z1} * G - z * K *Substitute Equation (4) in Equation (3)*

= z * r * G + f * G - z * G + z1 * G - z * K

= z * K + f * G - z * G + z1 * G - z * K

= f * G - z * G + z1 * G

Since z ≠ z1,

x-coordinate (W) ≠ x-coordinate (f * G)

v ≠ d

And Signature Verification fails

4. v = x-coordinate(W), finally, authenticate the signature by checking whether the equivalence v = d holds.

S = (z * r) + f mod n in an instance when the signature for the message m is (d, s) and was actually created by the authorized Sender. The aforementioned demonstration thus establishes that the ECDSA approach is effective in fending off the man-in-the-middle attack.

2.4.2 ***Replay Attacks***

To avoid replay attacks, both the sender and the recipient should create a completely random session key, which is a type of code that is only valid for one transaction and cannot be reused. Another safeguard against this kind of assault is the use of timestamps in all messages. This limits the window of opportunity for an attacker to eavesdrop, syphon out the message, and resent it by prohibiting hackers from resending communications transmitted after a particular period of time.

Sender: Bob Signature Generation

$$\begin{aligned} d &= x - coordinate\,(f * G) \\ s &= [\,(z * r) + f\ mod\ n] + Na \end{aligned} \tag{5}$$

Where Na is the Timestamp/Nonce added for the Signature Generation Session at the Sender Side. It is a random number for that session only

Receiver: Alice Signature Verification

$$W = (x1, y1) = s * G - z * K \tag{6}$$

Intruder: Darth Replay Attack
W = [((z * r) + f) + Na')] * G – z * K
Substitute Equation (5) in Equation (6)
Na' is time stamp created for this session and Na' ≠ Na
V = x-coordinate (W)
x-coordinate (W) ≠ x-coordinate (f * G)
Hence, v ≠ d
As Na' ≠ Na, doesn't match the time created at the Signature Verification session

2.4.3 ***The validation of the algorithm can be tested using the following proof for Replay Attack***

The following proof can be used to determine whether the algorithm is correct:
Replay Attack at the Signature Verification Side:

2.4.3.1 **Signature Generation Phase**. 1. d = x-co-ordinate (f * G)
2. s = [(z * r) + f mod n] + Na
Where Na is the Timestamp/Nonce added for the Signature Generation Session at the Sender Side. It is a random number for that session only

2.4.3.2 **Signature Verification Phase**. W = (x1, y1) = s * G – z * K
Substitute Equation (5) in Equation (6)
= [((z * r) + f) + Na')] * G – z * K
= [z * r + f + Na'] G – z * K
= [(z * r) * G + (f * G) + Na' * G – z * K
= z * K + f * G + Na' * G – z * K
= f * G +Na' * G
= (f + Na') * G
V = x-coordinate (W)
x-coordinate (W) ≠ x-coordinate (f * G)
Hence, v ≠ d
As Na' ≠ Na, does not match the time created at the Signature Verification session

2.5 Digital Forgery Attack

Digital signature forgery is the ability to create a message and a signature that are both valid but have never been created by the legitimate Signer. The Proposed Certificateless, Provably- Secure ECDSA method forbids the creation of counterfeit digital signatures.
Sender: Bob Signature Generation
(d, s) is the signature for the message m

$$s = [(z * r) + f \ mod \ n] \tag{7}$$

Intruder: Darth Forgery Attack
s'= fake signature

$$s' = (z' * r') + [((z' + p') \oplus (p' + r'))] \ mod \ n \tag{8}$$

Even though Darth avoids solving p', forging is impossible due to random r'.
The correctness of the algorithm can be tested using the following proof for Forgery Attack:
d: Private key of Sender
m: message
z: hash value of message m
r: random integer number in interval [0, n-1].

p = random integer p (secret key of Sender) with $1 \leq p \leq n-1$.
s = signature generated by Sender
$\oplus$ = Ex-OR operation
f = ((z + p) $\oplus$ (p + r))
s1= fake signature
Signature for the message m is (d, s)

2.5.1 *Fake Signature Generation*

Despite being unable to obtain the Signer's private key, if an attacker can get the Signature for the message m, it is (d, s).

The attacker then wants to forge the Signature.

$$\begin{aligned} &\bullet\ s = [(z * r) + f \bmod n] \\ &\bullet\ s1 = (z * r1) + [((z + p1) \oplus (p1 + r1))] \bmod n \end{aligned} \tag{9}$$

The attacker even though avoids solving p1, however because of randomness r1, forgery is out of question.

3 Results and Discussions

3.1 Input Specifications for ECDSA

The Weirstrass ECC curves are used for the experiment. The notations of the ECC curve are briefed below:

E: The elliptic curve under consideration, which is defined over the field GF(p) where p is a large prime and consisting of the point at infinity and the points (x, y) satisfying the equation

E: $y^2 = x^3 + ax + b \pmod p$ where a and b are constants and $4a^3 + 27b^2 \neq 0 \pmod p$.

p: A large prime which specifies the field over which the elliptic curve is defined, GF(p).

a and b: Constant curve parameters

x and y: The x and y coordinates of an affine point on the curve.

G: A point on the curve with order n, referred to as the basepoint and forming part of the domain parameters.

P, Q and R: Points on the curve.

#E(GF(p)) or η: The number of points on the curve, also known as the order of the curve.

n: The large prime order of the group of elliptic curve pointsc: A value such that $\eta = \#E(GF(p)) = c \cdot n$.

d: The private key of a user of the curve such that $d \in [1, n-1]$.

W: The public key of a user of the curve. W is found using the equation W = [d]G.

$r \in R$ S: r is randomly chosen from the set S.

The NIST standards for ECC at official the website https://csrc.nist.gov are used for experiment analysis. The performance metrics of the Proposed Certificateless, Provably- Secure ECDSA are described by the standards at www.ietf.org.The parameters are represented in figure1 to figure 6.

1. keygen: Time taken to generate key pairs

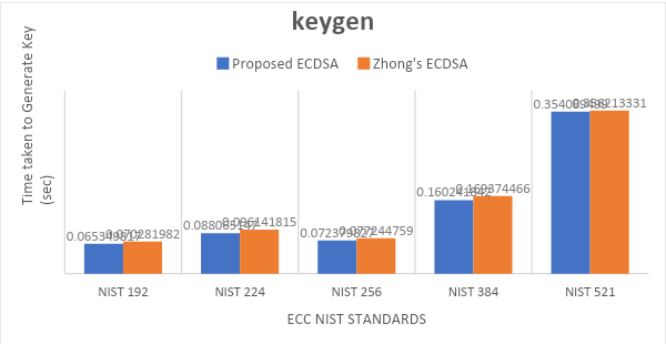

Fig 1. keygen for Proposed ECDSA and Zhong's ECDSA

The results in Figure 1 depict that the key generation time of the Proposed ECDSA using the standard NIST standards is 0.564 % less than Zhong's Method. The resultant values are an average of 10 cycles of execution with the standard NIST input

parameters.

2. keygen/s: How many keys per second can be generated

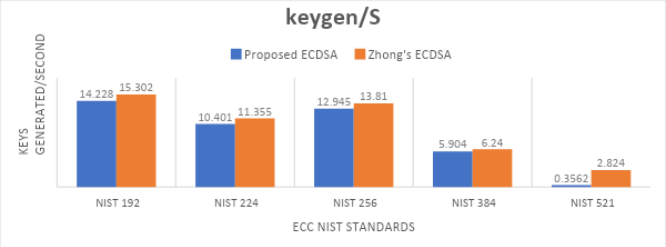

Fig 2. keygen/s for Proposed ECDSA and Zhong's ECDSA

The results in Figure 2 depict the number of keys generated/second by the Proposed ECDSA and Zhong's Method using the standard NIST standards. The Proposed method generates 1.1% lesser number of keys than Zhong's Method which is not a matter of concern for our scope as we are focused more on the time factor in real time applications. The resultant values are an average of 10 cycles of execution with the same input parameters.

3. sign: Time taken to sign data

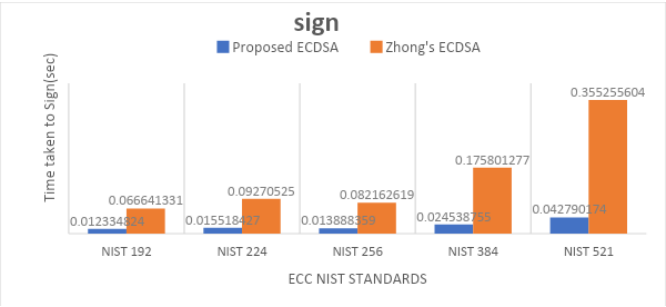

Fig 3. sign for Proposed ECDSA and Zhong's ECDSA

The results in Figure 3 depict that the time taken to sign data by the Proposed ECDSA using the standard NIST standards is considerably less than Zhong's Method. The Zhong's Method takes 13.28% more time to sign data than the Proposed ECDSA Method making our method more applicable in real time. This is a critical requirement of applications where communication is time critical. The resultant values are an average of 10 cycles of execution with the same input parameters.

4. **sign/s**: How many signatures can be made per second

The results in Figure 4 depict the number of signatures generated/second by the Proposed ECDSA and Zhong's Method using the standard NIST standards. The Proposed ECDSA method generates 47.15 % more number of signatures than Zhong's Method making it stand out in wider application areas. The resultant values are an average of 10 cycles of execution with the same input parameters.

5. **verify**: Time taken to verify signature

The results in Figure 5 depict the time taken to verify signature at the Receiver end by the Proposed ECDSA and Zhong's Method using the standard NIST standards. The resultant values are an average of 10 cycles of execution with the same input parameters. The Proposed ECDSA method takes 8.2% less time than Zhong's Method for Signature verification at the Receiver end making it stand out in wider application areas where computation time is of concern.

6. verify/s: How many signatures can be verified per second

The results in Figure 6 depict the number of signatures verified/second at the Receiver end by the Proposed ECDSA and Zhong's Method using the standard NIST standards. The resultant values are an average of 10 cycles of execution with the same input parameters. The Proposed method verifies 0.62% greater number of signatures than Zhong's Method.

Due to the wide range of applications in critical sectors, security is essential to the success of every internet application. Researchers have been using a variety of techniques for decades to create reliable digital signature systems that can withstand security flaws. By reducing the amount of elliptic curve mathematical operations, they are also attempting to lower the associated

Fig 4. sign/s for Proposed ECDSA and Zhong's ECDSA

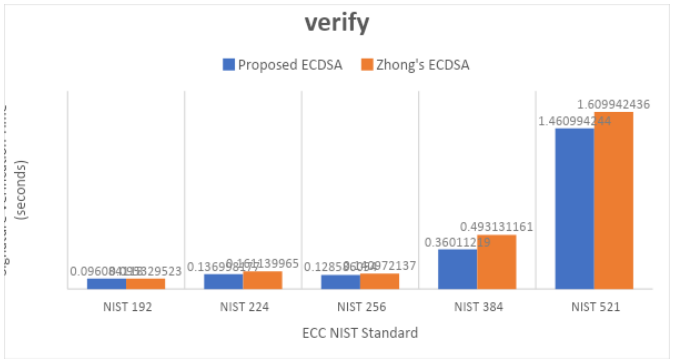

Fig 5. verify for Proposed ECDSA and Zhong's ECDSA

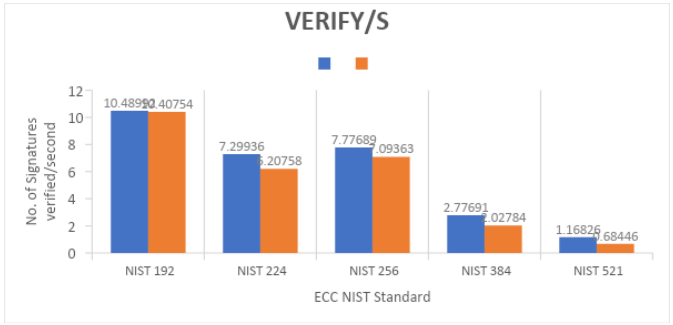

Fig 6. verify/s for Proposed ECDSA and Zhong's ECDSA

processing expenses. The systematic examination of several versions is evaluated for computing effort and security in terms of thwarting attacks.

3.2 Comparison of ECDSA Schemes w. r. t Resistance to Attacks

The Proposed Certificateless, Provably- Secure ECDSA Method is resistant to Replay attacks, MITM and forgery attacks as compared to Zhong's ECDSA Method which could sustain only replay attacks. Thus, the Proposed Certificateless, Provably-Secure ECDSA scheme without adding any overheads to computations or any need of any Certificate scheme to generate keys is sturdier. The Table 1 summarizes the resistance of the schemes to the attacks.

Table 1. Comparison of Proposed ECDSA & Zhong's ECDSA wrt Resistance to Attacks

SCHEME	Resistant to Attacks
Hong Zhong et al [15]	Replay Attack
Proposed Certificateless, Provably- Secure ECDSA	Replay, MITM Forgery

4 Conclusion

To determine the most significant Man in the Middle attack weakness, the security of Hong Zhong's plan is examined, and cryptanalysis is carried out. The cryptanalysis of Hong Zhong scheme attempts to achieve potency by decreasing the reserve standard inverse operations, but it fails to achieve security because an attacker can easily change the message and replace the current message's hash value with a different hash value, negating the scheme's attempts to meet the security requirements for a digital signature. The flaw in its peer Zhong's scheme is fixed by the suggested enhanced ECDSA scheme. The proposed Certificateless, Provably- Secure ECDSA outperforms Zhong's ECDSA as it is resistant to digital fabrication and attacks.

For key generation pairs, Zhong's Method requires 0.564% longer time than the proposed elliptical curve digital signature. The number of keys produced by the proposed ECDSA technique is 1.1% fewer than those produced by Zhong's method, however this is not relevant to our work because we are more concerned with the time factor in real-time applications. Our method is more relevant in real time since the Zhong's Method takes 13.28% less time to sign data than the suggested ECDSA method. The suggested technique stands out in broader application areas where calculation time is a concern since it requires 8.2% less time than Zhong's Method for Signature verification at the Receiver end.

References

1) Mahmoud AY. A Novel Hash Functions for Data Integrity Based on Affine Hill Cipher and Tensor Product. *International Journal of Engineering Trends and Technology*. 2022;70(11):1–9. Available from: https://doi.org/10.14445/22315381/IJETT-V70I11P201.
2) Gadde S, Amutharaj J, Usha S. A Hybrid Cryptography Technique for Cloud Data Security. *International Journal of Engineering Trends and Technology*. 2022;70(11):258–267. Available from: https://doi.org/10.14445/22315381/IJETT-V70I11P228.
3) Gattoju S, and VN. An efficient approach for bigdata security based on Hadoop system using cryptographic techniques. *Indian Journal of Computer Science and Engineering*. 2021;12(4):1027–1037. Available from: https://doi.org/10.21817/indjcse/2021/v12i4/211204132.
4) Bedoui M, Bouallegue B, Ahmed AM, Hamdi B, Machhout M, Mahmoud, et al. A Secure Hardware Implementation for Elliptic Curve Digital Signature Algorithm. *Computer Systems Science and Engineering*. 2023;44(3):2177–2193. Available from: https://doi.org/10.32604/csse.2023.026516.
5) Du H, Wen Q, Zhang S, Gao M. A new provably secure certificateless signature scheme for Internet of Things. *Ad Hoc Networks*. 2020;100:102074–102074. Available from: https://doi.org/10.1016/j.adhoc.2020.102074.
6) Noori D, Shakeri H, Torshiz MN. Scalable, efficient, and secure RFID with elliptic curve cryptosystem for Internet of Things in healthcare environment. *EURASIP Journal on Information Security*. 2020;2020(13):1–11. Available from: https://doi.org/10.1186/s13635-020-00114-x.
7) Singh AK, Solanki A, Nayyar A, Qureshi B. Elliptic Curve Signcryption-Based Mutual Authentication Protocol for Smart Cards. *Applied Sciences*. 2020;10(22):8291–8291. Available from: https://doi.org/10.3390/app10228291.
8) Rao V, V PK. Light-weight hashing method for user authentication in Internet-of-Things. *Ad Hoc Networks*. 2019;89:97–106. Available from: https://doi.org/10.1016/j.adhoc.2019.03.003.
9) Ahmed AA, Barukab OM. Unforgeable Digital Signature Integrated into Lightweight Encryption Based on Effective ECDH for Cybersecurity Mechanism in Internet of Things. *Processes*. 2022;10(12):2631–2631. Available from: https://doi.org/10.3390/pr10122631.
10) Yang J, Fan J, Zhu X. Perception Layer Lightweight Certificateless Authentication Scheme for IoT-Based Emergency Logistics. *IEEE Access*. 2023;11:14350–14364. Available from: https://doi.org/10.1109/ACCESS.2023.3243624.
11) Jammula M, Vakamulla VM, Kondoju SK. Artificial intelligence framework-based ultra-lightweight communication protocol for prediction of attacks in <scp>Internet of Things</scp> environment. *Transactions on Emerging Telecommunications Technologies*. 2023;34(1):4680–4680. Available from: https://doi.org/10.1002/ett.4680.
12) Shah TA, Ullah I, Khan MA, Lorenz P, Innab N. An Efficient Certificateless Forward-Secure Signature Scheme for Secure Deployments of the Internet of Things. *Journal of Sensor and Actuator Networks*. 2023;12(1):10–10. Available from: https://doi.org/10.3390/jsan12010010.
13) Wang Z, Zhao J, Sun P, Yang J, Wang R, Zhang X. A Lightweight Three-Party Mutual Authentication Protocol for Internet of Health Things Systems. *Journal of Healthcare Engineering*. 2023;2023:1–15. Available from: https://doi.org/10.1155/2023/1044282.
14) Chow MC, Ma M. A Lightweight D2D Authentication Scheme against free-riding attacks in 5G cellular Network. In: Proceedings of the 2nd International Electronics Communication Conference. Association for Computing Machinery. 2020;p. 143–149. Available from: https://doi.org/10.1145/3409934.3409952.
15) Zhong H, Zhao R, Cui J, Jiang X, Gao J. An Improved ECDSA Scheme for Wireless Sensor Network. *International Journal of Future Generation Communication and Networking*. 2016;9(2):73–82. Available from: http://article.nadiapub.com/IJFGCN/vol9_no2/8.pdf.