AN ATTACK-RESISTANT LIGHT-WEIGHT DIGITAL SIGNATURE BASED ON AN ELLIPTIC CURVE FOR IMPROVED SECURITY IN RESOURCE CONSTRAINED APPLICATIONS

Dr. Dhanashree K Toradmalle
Associate Professor, Department of Computer Engineering,
K J Somaiya Institute of Technology, Sion, Mumbai-77,
e-mail: dhanashree.t@somaiya.edu

Dr. Amarendra K
Professor, Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation,
Vaddeswaram 522502, Andhra Pradesh, India
e-mail: amarendra@kluniversity.in

Abstract

Despite the fact that ECDSA is the most innovative asymmetric digital signature technique, experts are working tirelessly to strengthen it to survive various challenges. Both internal and external attacks can occur from intruders. An end-user, a malware-infected IT component, a physical attacker who operates within the environment's security perimeter, or a physical person who directly interacts with the environment, manages the hardware, or even communicates with the end-user (i.e., a malicious signer). In contrast, an external attack involves the attacker moving outside the signature environment's security boundary, possibly across a network. Attacks on interfaces place more emphasis on the protocols that a device employs to interact with the outside world rather than on the machine itself. The proposed work presents a solution to an improved, lightweight ECDSA which is resistant to MITM, Replay and forgery attacks than its counterparts. The comparison of the proposed ECDSA is compared with its counterpart and cryptanalysis is performed to prove that the proposed ECDSA 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.

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] in big data security [3] and wireless networks. Additionally, though ECDSA is the most innovative asymmetric digital signature technique, experts are making every effort to strengthen it so that it can withstand various challenges. The usage of an elliptic curve-based digital signature scheme is also suggested by scientists [4-5]. The following are the requirements to toughen the ECDSA where current advanced study is being conducted:

Advance Secrecy [6-7]: Digital signatures allow the Signer to ensure the security of messages that have already been marked, regardless of whether the Signer's enigmatic key is found today. Attacks on Security: A number of assaults, such as man-in-the-middle and replay attacks, put security at risk. The ECDSA algorithm is appropriate for usage in numerous WSN [8-9], RFID [10] and smart card [11] implementations due to its performance and security. ECDSA digital signatures are more effective than DSA and RSA ones in constrained-resource devices. Numerous writers have suggested utilizing ECDSA in resource-constrained situations (memory, energy, and CPU capability). The range of attacks on digital signature are:

1. Man-in-the-middle [12][13] For an attack to succeed in getting around authentication, interface access is necessary. It does this by intercepting and fabricating messages in a way that compels the device to communicate with the attacker using a key they know (Clark et al. 1996). The connection with and without MIM attack is demonstrated in Fig.1.4 and 1.5 respectively. The replay attack is the most well-known MIM strategy.

● Attacks that replay [14]: Replay attacks (Syverson 1994) involve the attacker storing communications and sending them at irregular intervals.

A third person, say Mike, can go in between the two parties, those who are using signatures for authentication. Let's consider an instance where Bob is the sender. After Bob adds the digital signature and she receives the message, Alice decrypts it using the public key. The recipient, who is playing the role of Bob, must decrypt the message using Mike's public key if Mike in the middle signed it with his private key. Only the fact that Bob is the owner of the public key in this instance is assured; Bob's public key's authenticity is not. There is no assurance that the message came from Bob alone, thus. The following Fig.1.1 and Fig.1.2 present an illustration

Fig.1.1 Connection without man-in-the-middle attack

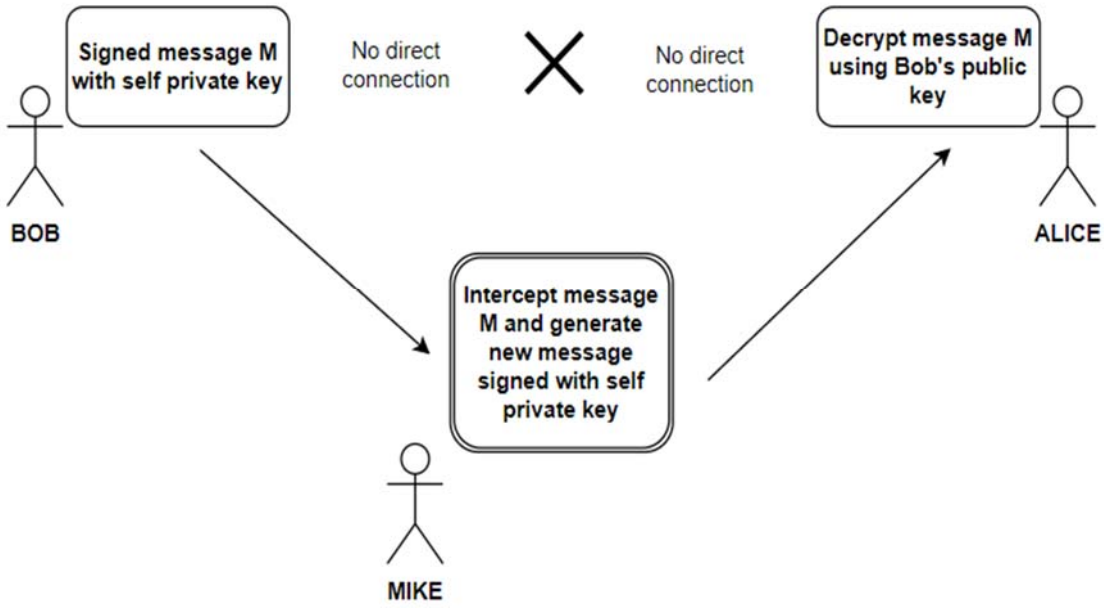

Fig.1.2 Connection with man-in-the-middle attack

Attack using a digital signature forgery: In a cryptographic digital signature or MAC system, a digital signature forgery is the capacity to construct a pair consisting of a message, m, and a signature, sigma, that is valid for m but has never been produced before by the legitimate signer.

2. Literature Survey

It contains a listing of solutions to produce a more ECDSA. ECC [15] Solutions are widely accepted and deployed in resource constrained applications like wireless adhoc networks [15-17] and IoT [18-21]. Assaults, security requirements, and countermeasures should first be divided into physical and non-physical assaults, with each category comprising both passive and active strikes. Furthermore, to avoid manipulation with ECDSA signatures, security measures are proposed. Many techniques are devised to withstand attacks to improve the security issues in networks [22]. Attacks are divided into several categories. Attacks are growing more sophisticated, thus there

should be countermeasures in place to protect against them. However, in terms of time, storage, and sophisticated computations, these defenses are too expensive. Furthermore, developing a countermeasure for each attack is difficult [23]. Countermeasures against known attacks were required when the ECDSA method was developed [24]. Use dependable standard criteria from bodies like IEEE, ISO, NIST, NSA, FIPS, and ANSI to prevent a broad range of attacks. The ECC/ECDSA algorithm's private key (d) and ephemeral key (k) protection methods are critical because if an opponent obtains the private key (d) and ephemeral key (k), the algorithm would be compromised. As a result, a set of countermeasures has been implemented to enhance the security procedures. Non-physical and physical assaults will be classified using the ECDSA method. With the aim of breaching message signatures' integrity, authenticity, and non-repudiation [25], many attacks fall under each of these categories.

2.1 Non-Physical Attacks and Security Requirements

Such attacks can access data repositories or communications sent between clients and servers without needing direct or indirect access to a client's computer or network. The frequent use of direct assaults [26] to decrypt signatures and encoders conveyed by radio frequency signals or the Internet is shown in Fig. 2.1. These attacks include sniffing, spoofing, eavesdropping, and manipulation. During the analysis and modification process, they avoid the security criteria of confidentiality, integrity, and availability (CIA). Network hardware may be indirectly employed in cyberattacks. They can be classified as passive or active attacks depending on the methods employed and the targeted target.

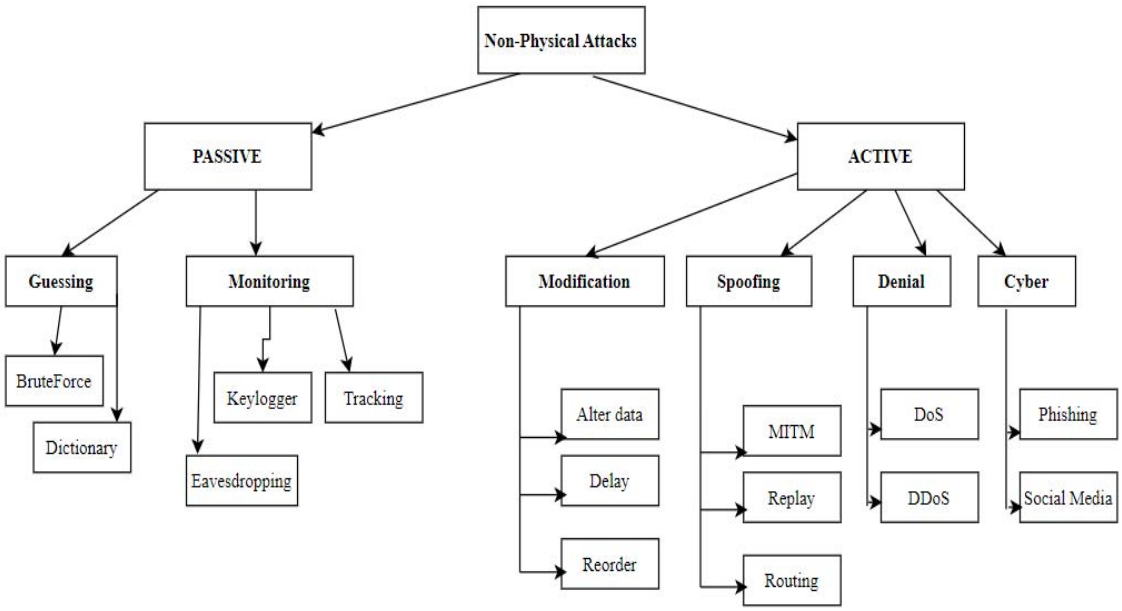

Fig.2.1. Classification of non-physical attacks [83]

The security and non-repudiation features pertaining to ECDSA are put at risk by attacks. By concentrating only on the modular operations, Xianmin Wei et al., [27] argue that ECC has improved but fall short of capturing the security aspects of assaults. Neetesh Saxena et al., [28] present ECDSA variants that emphasize efficiency while underestimating security attacks. Chang et al., cryptanalysis is demonstrated by Jie Liu et al., [29]. The claim that the digital signature system was resistant to forging attacks was made without the use of one-way hashing or padding with redundant data. They also suggest enhanced signature schemes, in which the digital signature is significantly reduced in length. Lei Niu et al.,[30] describes a series of forgeries attacks in order to demonstrate the most effective method of obtaining the attacks. Jianhong Zhang et al.,[31] deconstruct blind signatures and show that they are inherently insecure, before proposing a remedy. An enhanced technique for avoiding forgery attacks is provided by Xinghua Zhang et al.,[32]. The "Man-in-the-Middle" attacks are limited by Long Zhaohua et al., [33], but the application overhead is increased because three entities, such as Station (STA), Access Point (AP), and Authentication Server (AS), must take part in the wireless network's identity authentication process. To reduce the computational expense required during the process of creating and verifying signatures, Hong Jhong's method [34] makes an effort to achieve strength by omitting the inverse standard operations, however it is unable to ensure security.

Cryptology continually attempts to fill weaknesses in information correspondence made by the attackers, who continuously endeavor to break the signature calculations [35]. Hence, it is integral to withstand these attackers.

These intruders attempt to enter the security limits leading to exposure of security flaws and they are successful in their damage work. The research work is based on the motivation below

- It is crucial to constantly reenergize and update the digital signatures in order to thwart attempts to access secret information.
- By developing novel digital signature schemes based on elliptic curves, security targets can be met, such as confidentiality, legitimacy, and non-revocation.

The challenges with ECDSA are the computational time and speed. These parameters are dependent on the number of ECC operations. The issue for security is to develop solutions that meet the client's urgent demand in resource-constrained settings at a lower machine speed and cost. While looking for alternatives, researchers are keeping in mind ECDSA's requirements, which include strong security and reduced key sizes.

3. 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 [34] 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 because it is vulnerable to a hacker completely changing the message and replacing the current message hash value with a different hash value. ECDSA inherits from ECC the advantages of a small key size and good security. An enhanced approach of ECDSA was proposed by Hong Zhong et al. The notations used are as follows:

The notations used are as follows:

G: Basepoint of elliptic curve

d: Private key of Alice m: message

e: hash value of message m

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)

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.

3.1 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.

4. Proposed Attack-resistant ECDSA

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.

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)

Signature Verification Phase
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.

4.1 Security Proof of the proposed ECDSA

4.1.1 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:

W = s * G – z * K = ((z * r) + f) * G – z * K (Eqn.4.1)
= z * r * G + f * G – z * K
= z * K + f * G – z * K
= f * G x-coordinate (W)
= x-coordinate (f * G)

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 demonstrated in Fig.4.1, which is defeated by the evidence proposed above.

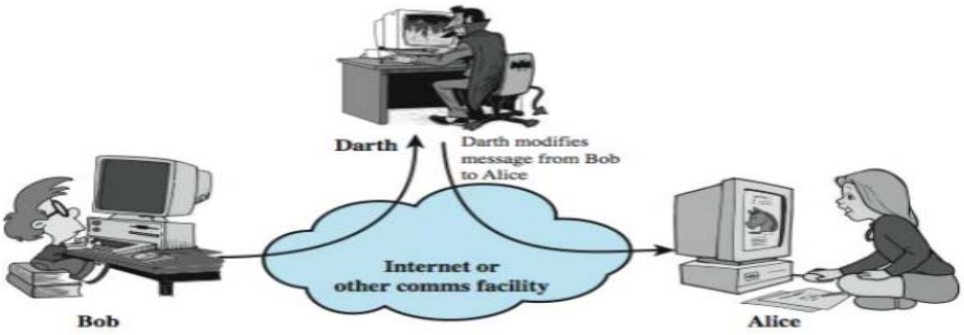

Fig.4.1 MITM Attack

Sender: Bob Signature Generation

$s = [(z * r) + f \bmod n]$ (Eqn.4.2)

Receiver: Alice Signature Verification

$W = (x1, y1) = s * G – z * K$ (Eqn.4.3)

Intruder: Darth MITM Attack

$s1 = [(z * r) + f \bmod n] - z + z1$ (Eqn.4.4)

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

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. $W = (x1, y1) = s * G – z * K$

$W= \{[(z * r) + f \bmod n] -z +z1\} * G - z * K$ *Substitute Eqn 4.4 in Eqn 4.3*

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

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

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

Since $z \neq z1$,

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

$v \neq d$

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.

4.1.2 Replay Attacks

When a hacker listens in on a secure network connection, intercepts it, and then falsely delays or resends it to the recipient to coerce them into doing what the hacker wants, this is called a replay attack [36], as shown in Figure.4.2. Replay attacks represent an additional risk because, after obtaining a message from the network, a hacker doesn't even need advanced skills to decrypt it. The attack could succeed by simply broadcasting the full thing again. To avoid this scenario, 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.

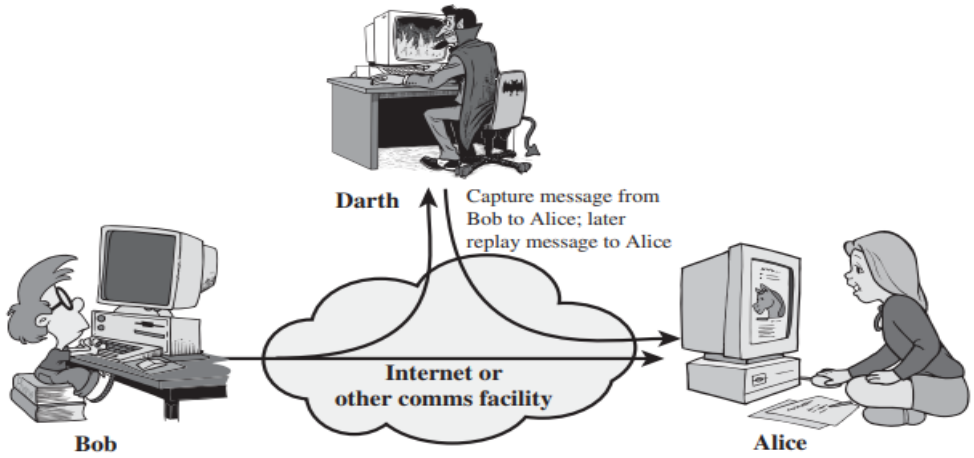

Fig.4.2 Replay Attack

Sender: Bob Signature Generation

d = x-coordinate (f * G)

s = [(z * r) + f mod n] + Na (Eqn.4.5)

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 (Eqn.4.6)

Intruder: Darth Replay Attack

W = [((z * r) + f) + Na')] * G – z * K *Substitute Eqn.4.5 in Eqn.4.6*

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

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:

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

Signature Verification Phase

W = (x1, y1) = s * G – z * K

Substitute Eqn.4.5 in Eqn. in 4.6

= [((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

4.1.3 Digital Forgery Attack

Digital signature forgery, illustrated in Fig. 4.3, is the ability to create a message and a signature that are both valid but have never been created by the legitimate Signer. The suggested ECDSA method forbids the creation of counterfeit digital signatures.

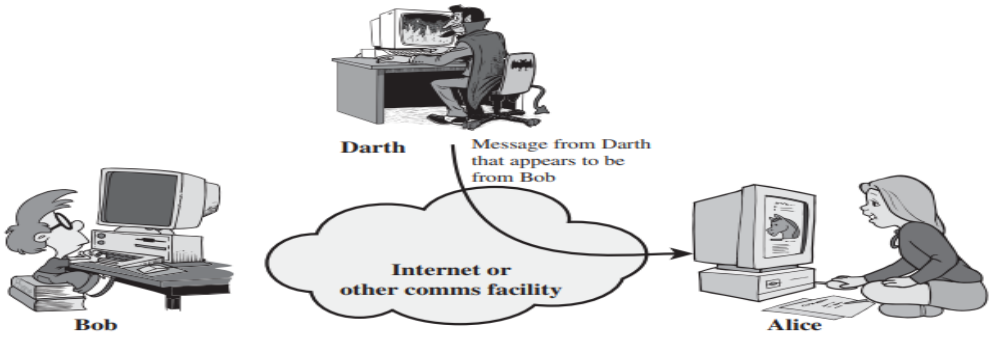

Fig.4.3 Forgery Attack

Sender: Bob Signature Generation

(d, s) is the signature for the message m

s = [(z * r) + f mod n] (Eqn.4.7)

Intruder: Darth Forgery Attack

s'= fake signature

s'= (z' * r') + [((z' + p') ⊕ (p' + r'))] mod n (Eqn.4.8)

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
⊕ = Ex-OR operation

f = ((z + p) ⊕ (p + r))

s1= fake signature

Signature for the message m is (d, s)

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.

- s = [(z * r) +f mod n]
- s1= (z * r1) + [((z + p1) $\oplus$ (p1 + r1))] mod n (Eqn.4.9)

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

5. 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: $y^2 = x^3 + ax + b$ (mod p) where a and b are constants and $4a^3 + 27b^2 \neq 0$ (mod 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 points
c : 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 [37] are used.

The performance metrics [38] of the Proposed ECDSA are listed in Table 5.1 below.

Sr.No	Performance metrics	Description
1	Signlen	Signature Length
2	Keygen	Time taken to generate key pairs
3	keygen/s	How many keys per second can be generated
4	Sign	Time taken to sign data
5	sign/s	How many signatures can be made per second
6	Verify	Time taken to verify signature
7	verify/s	How many signatures can be verified per second

Table.5.1. Performance metrics for ECDSA

6. Results and Discussion

1. keygen: Time taken to generate key pairs

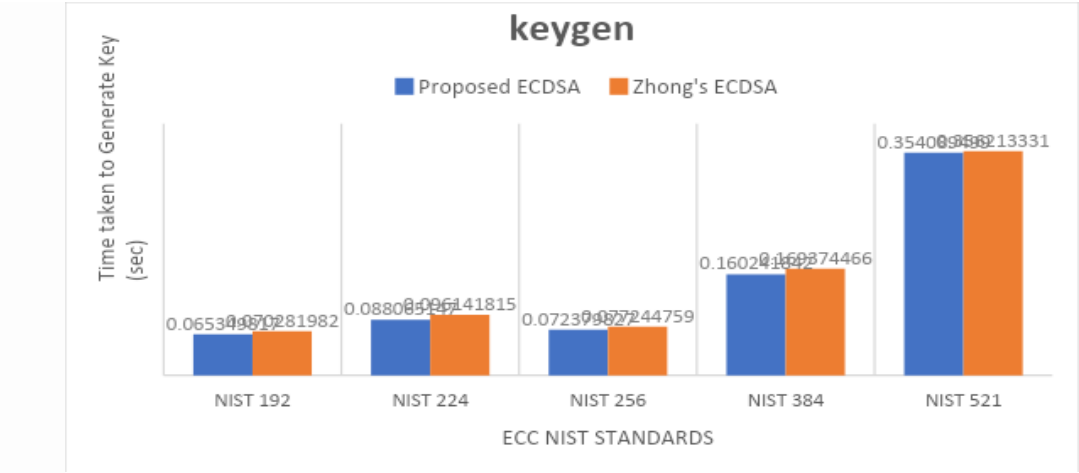

Fig.6.1 keygen for ECC NIST Standards

The results in Fig.6.1 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

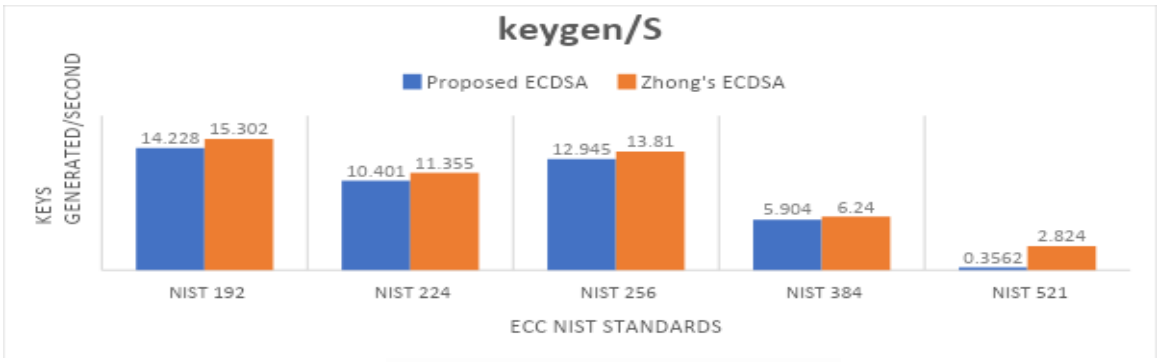

Fig.6.2 keygen/s for Proposed ECDSA and Zhong's ECDSA

The results in Fig.6.2 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.6.3 sign for Proposed ECDSA and Zhong's ECDSA

The results in Fig.6.3 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

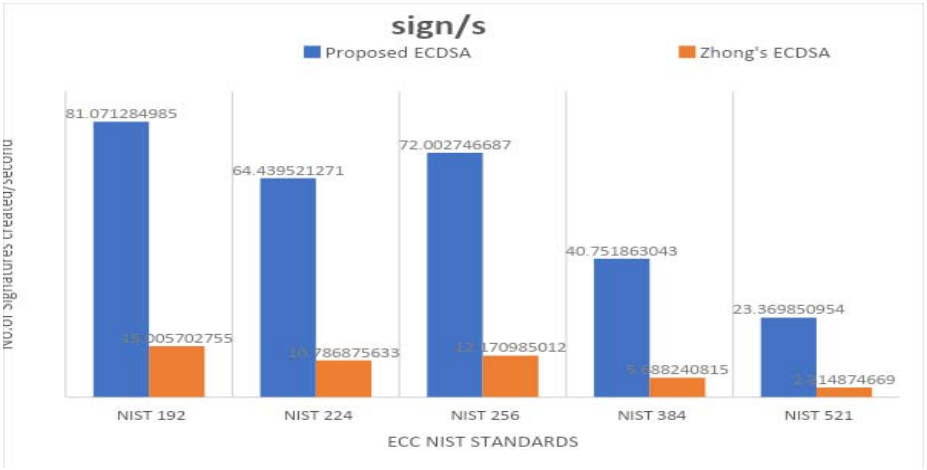

Fig.6.4 sign/s for Proposed ECDSA and Zhong's ECDSA

The results in Fig.6.4 depict the number of signatures generated/second by the Proposed ECDSA and Zhong's Method using the standard NIST standards. The Proposed 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

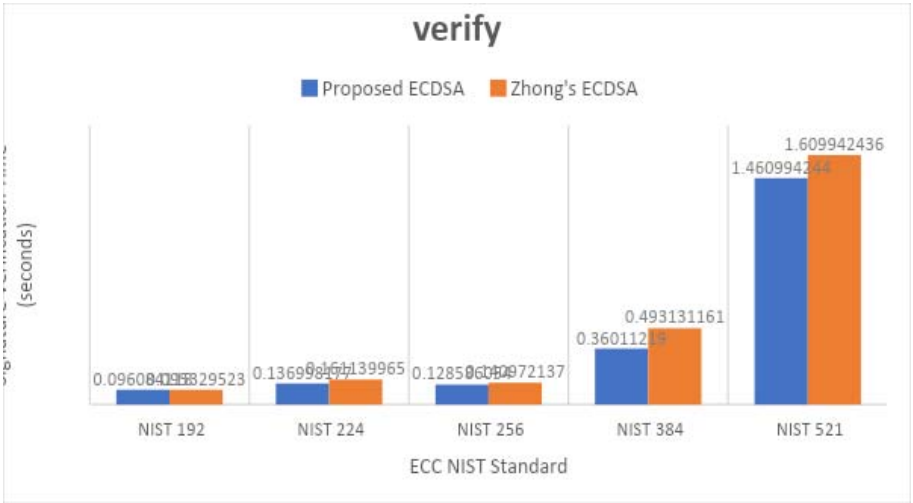

Fig.6.5 verify for Proposed ECDSA and Zhong's ECDSA

The results in Fig.6.5 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 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

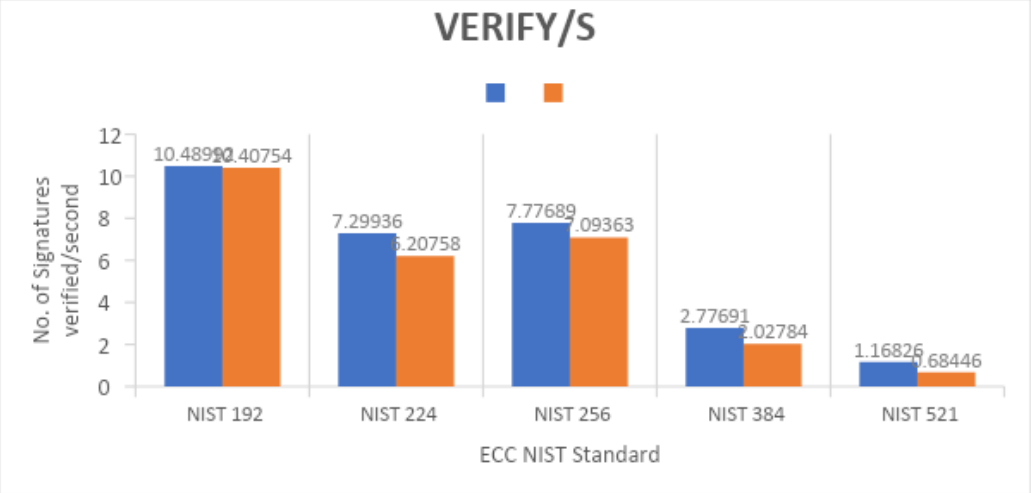

Fig.6.6 verify/s for Proposed ECDSA and Zhong's ECDSA

The results in Fig.6.6 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 processing expenses. The systematic examination of several versions is evaluated for computing effort and security in terms of thwarting attacks.

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

The Proposed 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 ECDSA scheme without adding any overheads to computations or any need of any Certificate scheme is sturdier. The Table 6.1 summarizes the resistance of the schemes to the attacks.

SCHEME	Resistant to Attacks
Hong Zhong et al [34]	Replay Attack
PROPOSED SCHEME	Replay, MITM Forgery

Table.6.1. Comparison of Proposed ECDSA & Zhong's ECDSA wrt Resistance to Attacks

Public encryption and digital signature functionality are combined into one step by a cryptographic method called Signcryption. Since it combines these two, it helps to achieve privacy, reliability, validation, and non-renunciation while doing so. The proposed approach is a certificateless system that is implemented taking into account the necessary conditions at minimal computation costs in order to make it generally applicable in locations with limited resources.

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 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 suggested system outperforms Zhong's, which is weak against man-in-the-middle attacks. Moreover, it is resistant to digital fabrication and replay assaults. In comparison to other variations, proposed ECDSA performs better in terms of the computational cost associated with signature generation, verification, or resistance to assaults. So, when compared to its competitors, the benefits of the suggested strategy make it stand out.

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] Ahmed Y. Mahmoud, "A Novel Hash Functions for Data Integrity Based on Affine Hill Cipher and Tensor Product," *International Journal of Engineering Trends and Technology*, vol. 70, no. 11, pp. 1-9, 2022.

[2] Swetha Gadde, J. Amutharaj, S. Usha, "A Hybrid Cryptography Technique for Cloud Data Security," *International Journal of Engineering Trends and Technology*, vol. 70, no. 11, pp. 258-267, 2022. https://doi.org/10.14445/22315381/IJETT-V70I11P228

[3] Saritha Gattoju, Dr. V. Nagalakshmi "An efficient approach for bigdata security based on Hadoop system using cryptographic techniques" Indian Journal of Computer Science and Engineering, Volume 12, Issue 4 Jul-Aug 2021 doi: https://doi.org/10.21817/indjcse/2021/v12i4/211204132.

[4] Jianglang Feng; Jindong Li, "A New Certificate Based Digital Signature Scheme", 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies

[5] Chae Hoon Lim, PilJoong Lee, "A Study on the Proposed Korean Digital Signature Algorithm", in ASIACRYPT '98 Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology, 2000, pp. 175-186

[6] B.B. Amberker, Prashant Koulgi, N.R. Sunitha, "Modified Forward Secure Signatures for Mobile Computing Applications," International Conference on wireless and optical communications network, IEEE, 2006.

[7] . Hong Jingxin, "A New Forward-Secure Digital Signature Scheme," International Workshop on Anti-Counterfeiting Security and Identification, IEEE, 2007.

[8] Akansha Singh, Amit K. Awasthi ,Karan Singh A Key Agreement Algorithm Based on ECDSA for Wireless Sensor Network, Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics pp 143–149.

[9] Lavanya M, Natarajan, V. LWDSA: light-weight digital signature algorithm for wireless sensor networks,2017, pp1629–1643.

[10] Pessl, P., Hutter, M. (2014). Curved Tags "A Low-Resource ECDSA Implementation Tailored for RFID" In: Saxena, N., Sadeghi, AR. (eds) Radio Frequency Identification: Security and Privacy Issues. RFIDSec 2015. Lecture Notes in Computer Science, vol 8651. Springer.

[11] Arazi, B. "Communication Computation Trade-off in Executing ECDSA in a Contactless Smartcard" *Des Codes Crypt*, 2006, pp 399–415.

[12] Defiana Arnaldy, Audhika Rahmat Perdana, "Implementation and Analysis of Penetration Techniques Using the Man-In-The-Middle Attack", 2019 2nd International Conference of Computer and Informatics Engineering (IC2IE).

[13] Sajal Jain, ShivamSharma,B. R. Chandavarkar, "Mitigating Man-in-the-Middle Attack in Digital Signature",11th ICCCNT IEEE 2020

[14] Jorge L. Hernandez-Ardieta, Ana I. Gonzalez-Tablas, Jose M. de Fuentes, Benjamin Ramos, "A taxonomy and survey of attacks on digital signatures",Elseveir Computers & Security Volume 34, May 2013, pp 67-112.

[15] Dr. V. Gokula Krishnan, Dr. J. Deepa, Dr. S. Venkata Lakshmi, T. A. Mohana Prakash, Dr. K. Sreerama Murthy, V. Divya, "Securing Mass Distributed Big Data Storage using Intelligent Elliptic Curve Integrated Encryption Scheme in Multi-Cloud Computing," *International Journal of Engineering Trends and Technology*, vol. 70, no. 3, pp. 29-36, 2022. https://doi.org/10.14445/22315381/IJETT-V70I1P204

[16] Y. Akshatha, A. S. Poornima, M. B. Nirmala, "Secure Data Collection in Clustered Wireless Sensor Networks using Fuzzy based scheme to detect Malicious Data Collector," *International Journal of Engineering Trends and Technology*, vol. 70, no. 11, pp. 240-248, 2022. https://doi.org/10.14445/22315381/IJETT-V70I11P226

[17] B. Murugeshwari, R. Amirthavalli, C. Bharathi Sri, S. Neelavathy Pari, "Hybrid Key Authentication Scheme for Privacy over Adhoc Communication," *International Journal of Engineering Trends and Technology*, vol. 70, no. 10, pp. 18-26, 2022. https://doi.org/10.14445/22315381/IJETT-V70I10P203

[18] Shaji K.A.Theodore, K. Rajiv Gandhi, V. Palanisamy , "Privacy Preserving Lightweight Cryptography Scheme for Clustered Vehicular Adhoc Networks" International *Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 24-31, 2022. https://doi.org/10.14445/22315381/IJETT-V70I7P203

[19] Srabana Pramanik, Deepak. S. Sakkari, Sudip Pramanik, "Remediation Measures to Make the Insecure Internet of Things Deployment Secure," *International Journal of Engineering Trends and Technology*, vol. 70, no. 6, pp. 155-164, 2022. https://doi.org/10.14445/22315381/IJETT-V70I6P219

[20] Rajat Verma, Namrata Dhanda, Vishal Nagar, "Enhancing & Optimizing Security of IoT Systems using Different Components of Industry 4.0." *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 147-157, 2022. https://doi.org/10.14445/22315381/IJETT-V70I7P216

[21] Phani Sridhar Addepalli, Dr.P.V.Lakshmi, "A hybrid security framework for medical data in IoT applications ", *Indian Journal of Computer Science and Engineering,* Volume 12 ,Issue 4 Jul-Aug 2021, https://doi.org/10.21817/indjcse/2022/v13i2/221302010

[22] Gang Xu, Allemar Jhone P. Delima, Ivy Kim D. Machica, Jan Carlo T. Arroyo, Zhengfang He, Weibin Su, "Improvement of Wireless Sensor Networks Against Service Attacks Based on Machine Learning," *International Journal of Engineering Trends and Technology*, vol. 70, no. 5, pp. 74-79, 2022. https://doi.org/10.14445/22315381/IJETT-V70I5P209

[23] C. Giraud and H. Thiebeauld, \A survey on fault attacks," in Smart Card Research and Advanced Applications VI. Springer, 2004, pp. 159-176.
[24] M. Varchola, M. Drutarovsky, M. Repka, and P. Zajac, “Side channel attack on multiprecision multiplier used in protected ecdsa implementation," in 2015 International Conference on ReConFigurable Computing and FPGAs (ReCon-Fig). IEEE, 2015.
[25] G. McGraw, Software Security: Building Security In. Addison-Wesley Professional, 2006.
[26] S.-R. Oh and Y.-G. Kim, Security requirements analysis for the iot," in Platform Technology and Service (PlatCon), 2017 International Conference on. IEEE, 2017
[27] Xianmin Wei, Peng Zhang, “Research on Improved ECC Algorithm in network and Information Security” International Journal of Security and Its Applications Vol.9, No.2 (2015), pp.29-36
[28] Neetesh Saxena, Narendra S. Chaudhari, Jaya Thomas, “Solution to an Attack on Digital Signature in SMS Security”, International Conference on Modeling, Simulation and Applied Optimization,2013 IEEE
[29] Jie Liu and Jianhua Li “Cryptanalysis and Improvement on a Digital Signature Scheme without using One-way Hash and Message Redundancy”, International Conference on Information Security and Assurance, IEEE 2008.
[30] Lei Niu, Yong Yu, Jianbing Ni, Ying Sun, “Further Cryptanalysis of a Signature Scheme with Message Recovery” International Conference on Intelligent Networking and Collaborative Systems, IEEE 2012.
[31] Jianhong Zhang, Shengnan Gao, “Cryptoanlaysis of a Self-Certified Partially Blind Signature and a Proxy Blind Signature”, WASE International Conference on Information Engineering, IEEE 2009
[32] Xinghua Zhang, “Security Analysis and The Improvement of the sequential multi-signature scheme based on discrete logarithm”, International Conference on Consumer Electronics Communications and Networks,2013 IEEE
[33] Long Zhaohua, Wang Guofeng, "Multi-element Authentication method based on ECDA for Wireless Network," International Symposium on Knowledge Acquisition and Modeling Workshop, IEEE, 2008
[34] Hong Zhong, Rongwen Zhao, Jie Cui*, Xinghe Jiang and Jing Gao, “An Improved ECDSA Scheme for Wireless Sensor Network”, International Journal of Future Generation Communication and Networking Vol. 9, No. 2 2016, pp.73-82.
[35] Hansman, S, Hunt R. A taxonomy of network and computer attacks. Computer and Security, 2005.
[36] M. Varchola, M. Drutarovsky, M. Repka, and P. Zajac, “Side channel attack on multiprecision multiplier used in protected ECDSA implementation," in 2015 International Conference on ReConFigurable Computing and FPGAs (ReCon-Fig). IEEE, 2015.
[37] https://csrc.nist.gov/csrc/media/publications/fips/186/2/archive/2000-01 27/documents/fips186-2.pdf
[38] https://www.ietf.org/proceedings/92/slides/slides-92-lwig-3.pdf

Authors Profile

Dr. Dhanashree K Toradmalle is working as an Associate Professor in the Department of Computer Engineering at K J Somaiya Institute of Technology, Sion, Mumbai. Her research areas include Computer Networks, Information and Network Security, Cyber security.

Dr. Amarendra K is currently working as Professor & HoD, Dept of CSE & IT, KLEF, Vijayawada,Andhra Pradesh. He is an experienced Professor with a demonstrated history of working in the Teaching Industry. Skilled in Computer Science lecturing C / C++ Programming, Data Structures, Algorithms, Software Engineering, Cyber Laws and Security