International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 3, June 2016, pp. 1002 ~ 1010
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i3.9420

Secure Digital Signature Scheme Based on Elliptic Curves for Internet of Things

Sumanth Koppula, **Jayabhaskar Muthukuru**
Department of Computer Science and Engineering, K L University

Article Info

Article history:

Received Nov 15, 2015
Revised Mar 3, 2016
Accepted Mar 15, 2016

Keyword:

Internet of things
Elliptic curve digital signature
Digital signature
Elliptic curves

ABSTRACT

Advances in the info and communication knowledge have led to the emergence of Internet of things (IoT). Internet of things (loT) is worthwhile to members, trade, and society seeing that it generates a broad range of services by interconnecting numerous devices and information objects. Throughout the interactions among the many ubiquitous things, security problems emerge as noteworthy, and it is significant to set up more suitable solution for security protection. Nonetheless, as loT devices have limited resource constraints to appoint strong protection mechanisms, they are vulnerable to sophisticated security attacks. For this reason, a sensible authentication mechanism that considers each useful resource constraints and safety is required. Our proposed scheme uses the standards of Elliptic Curve digital signature scheme and evaluates systematically the efficiency of our scheme and observes that our scheme with a smaller key size and lesser infrastructure performs on par with the prevailing schemes without compromising the security level.

Corresponding Author:

Sumanth Koppula,
Department of Computer Science and Engineering,
K L University,
Vaddeswaram 522502, Guntur District, Andhra Pradesh, India.
Email: koppulasumanth@live.com

1. INTRODUCTION

Advances in the wired, wireless, cellular and sensor networks have left a pretty good base for the internet of things (IoT). It's a novel paradigm which takes account of every day physical world objects through enabling interplay among them via targeted addressing schemes. Internet of things (IoT) refers back to the network interconnection of every day devices. An IoT is a global-vast community of inter-linked devices uniquely addressable, headquartered on a usual communication protocol. In the IoT, persons are bounded by utilizing one-of-a-kind forms of computing items which might be billion in number, various in size, and capabilities to remain a communication with each other device. It is predictable that around 50 billion such objects will be interconnected to the internet by means of 2020.These Devices are having constrained capabilities, and calculating resources ranges from Radio Frequency Identification (RFID) tags to embedded instruments, PDA, and sensor nodes. IoT joins the physical world with the information world, and presents ambient offerings, and applications. The IoT networks permits users, devices, and purposes in unique physical places to keep up a correspondence seamlessly with one another. Briefly, the IoT allows exceptional verbal exchange patterns like: person-to-person, person-to-object, object-to-object, and object-to-person. Still, the decentralized and dispensed nature of the IoT face challenges in authentication, entry control, and identification management. There are more than a few challenges to design protection options within the IoT like constraints, and heterogeneous conversation, resource constraints, and dispensed nature. Identity management of devices in the IoT is likely one of the primary task, and can be completed by using effective authentication schemes that are easy, secure, and lightweight. In the IoT, there are abundant

numbers of heterogeneous things chatting to each other. Every single device will have to be able to authenticate for the period of the short time. Due to the size of economics, more than enormous quantities of things may just request authentication approval at the same time. To this intention, lightweight, scalable, and secure authentication scheme is essential with the intention to authenticate organizations of devices, and now not the individual devices to obtain comfortable group communication.

2. ELLIPTIC CURVE ARITHMETIC

Elliptic curve cryptography is based on the arithmetic of points on an elliptic curve [1],[2]. Elliptic curves are characterized by cubic equations alike to those used for computing the circumference of an ellipse. An elliptic curve E over a field K is defined by a equivalence [3]:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

Where a_1, a_2, a_3, a_4, $a_6 \in$ k and $\Delta \neq 0$, where Δ is defined as follows:

$$\Delta = -d^2_2\, d_8 — 8d^3_4 — 27d2_6 + 9d_2d_4d_6;$$

Where $d_2 = a1_2 + 4a_2$,
$d_4 = 2a_4 + a_1a_3$,
$d_6 = a^2_3 + 4a_6$ and
$d_8 = a^1_2a_6 + 4a_2a_6a_1a_3a_4 + a_2a^2_3a^2_4$

Set of all points (x, y), which fulfils the above equation, are the points on the elliptic curve. The quantity of points on an elliptic curve, n, is the order of elliptic curve, (#(E(F_p)). The set of points of E (F_p) composed with addition operation forms an abelian group with point at infinity, ∞ as the identity element. The Equality {1} is called as weierstrass equation. The condition $\Delta \neq 0$ ensures that the elliptic curve is plane, i.e, there are no points at which the curve has two or more divergent tangent lines.

If the field representative P is not equal to 2 or 3 i.e., prime field, and then the permissible change of Variables (x, y) $\rightarrow$((x-3a_1^2-12a_2)/36, (y-3a_1x)/216- (a_1^3+4a_1a_2-12a_3)/36) transform E to the curve,

$$Y^2 = x^3 + ax + b; \text{ where } a, b \in k \quad (2)$$

The Δ is $16(4a^3 + 27b^2)$

2.1. Point Addition

Totaling of points on an elliptic curve is defined by Chord and Tangent rule. Let P = (x_1, y_1) and Q = (x_2, y_2) be the two dissimilar points on an elliptic curve E. Then the sum R, of P and Q, is defined as follows: Draw a line attaching P and Q spread it to intersect the elliptic curve at a third point. At that point, the sum R, is the negative of the third point. Negative of a point is defined by reflection of the point near the x-axis. The double R, of P, is defined as follows: Draw the tangent line to the elliptic curve at P. Let it interconnects the elliptic curve at another point. Then the double R is the reflection of this point near the x-axis.

2.2. Point Multiplication

It is also known as Scalar multiplication. It the arithmetic operation which calculates kp where k is an integer and p is a point on elliptic curve. It is completed by repetitive addition. For instance Q = kp means Q is achieved by adding p*k times to itself (p + p + p....k times). Cryptanalysis involves determining k given P and Q. This procedure dominates the implementation time of elliptic curve cryptographic schemes.

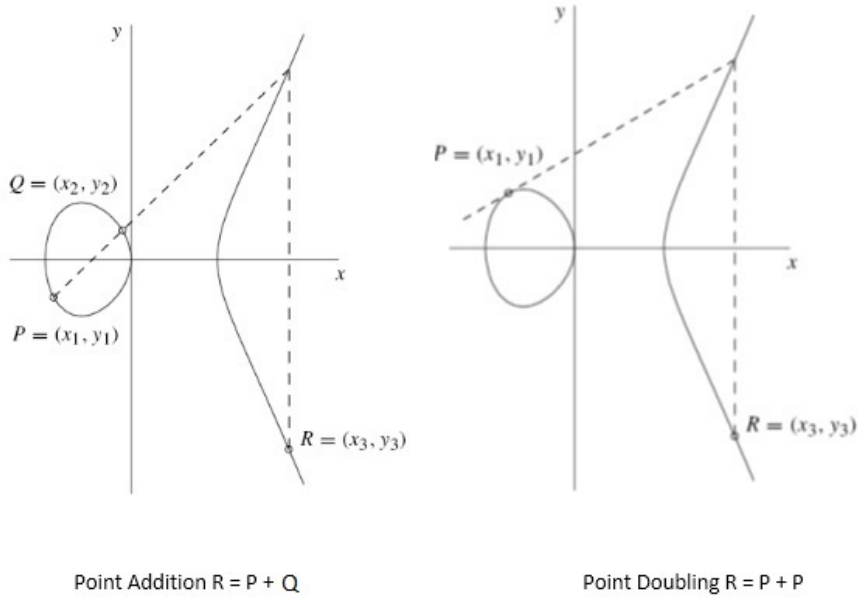

Figure 1. The implementation time of elliptic curve cryptographic schemes

2.3. Operations defined for $E(F_p)$: $y^2 = x^3+ax+b$

1. **Identity:** $P + \infty = \infty + P = P$ for all $>\in E(F_p)$
2. **Negatives:** If $P = (x,y) \in E(F_P)$, then $(x,y) + (x, -y) = \infty$. The point $(x, -y)$ is denoted by $-P$ and is called negative of P. Note that P indeed is a point in $E(F_P)$.
3. **Point Addition:** Let $P=(x_1,y_1) \in E(K)$ and $Q=(x_2,y_2) \in E(K)$; where $P \neq \pm Q$, then $P+Q = (x_3,y_3)$ where, $x_3 = (y_2 - y_1 / x_2 - x_1)^2 - x_1 - x_2$ and $y_3 = (y_2 - y_1 / x_2 - x_1)^2 (x_1 - x_3) - y_1$
4. **Point Doubling:** Let $P=(x_1,y_1) \in E(K)$, then $2P = (x_3,y_3)$ where, $x_3 = (3x_1^2 + a / 2\, y_1)^2 - 2x_1$ and $y_3 = (3x_1^2 + a / 2\, y_1)^2 (x_1 - x_3) - y_1$

2.4. Elliptic Curve Discrete logarithm problem

Assumed elliptic curve parameters and a point $P \in E(F_p)$, find the unique integer k, $0 \leq k \leq$ n1, such that P=kG, where n1 is order of E. ECDLP is alike to the Discrete Logarithm Problem and is the elliptic curve referent of DLP. In the ECDLP, the subgroup Z_p^* is altered by the group of points on an elliptic curve over a finite field. In addition, unlike the Discrete Logarithm Problem and integer factorization problem, no sub exponential-time algorithm is known for the ECDLP. ECDLP is considered to be significantly stronger than DLP, therefore elliptic curve signature scheme gives a greater strength-per-key-bit than their discrete logarithmic counterparts.

3. ELLIPTIC CURVE CRYPTOGRAPHY

The usage of Elliptic Curve Cryptography was primarily advised by Neal Koblitz [4] and Victor S. Miller [5]. Elliptic curve cryptosystems over finite field have some benefits like the key size can be considerably smaller compared to additional cryptosystems like RSA, Diffie-Hellman since only exponential-time attack is known so far if the curve is carefully chosen [4],[6] and Elliptic Curve Cryptography depend on the difficulty of explaining the Elliptic Curve Discrete Logarithm Problem ECDLP, which states that, "Given an elliptic curve E well-defined on a finite field F_P , a point $P \in E (F_P)$ of an order n, and a point $Q \in E (F_P)$, find the integer $k \in [0,n-1]$ such that Q = k P. The integer k is named as the discrete logarithm of Q to the base P, denoted $k = \log_P Q$".

3.1. Elliptic Curve Encryption/Decryption

Consider a message 'Pm' directed from A to B. 'A' picks a random positive integer 'k', a private key 'nA' and produces the public key $P_A = n_A \times G$ and produces the cipher text 'Cm' be made up of pair of points $Cm = \{kG, P_m + kP_B\}$ where G is the base point selected on the Elliptic Curve, $P_B = n_B \times G$ is the public key of B with private key 'nB'. To decrypt the cipher text, B reproduces the 1st point in the pair by B's secret & deducts the result from the 2nd point $P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$.

4. DIGITAL SIGNATURE SCHEMES

Digital signatures are being used to attain integrity, non-repudiation and authentication of the digital data in transmission among dissimilar end users. Digital signature offers correct architecture for sending secure messages by way of utilizing exceptional algorithms. The digital signature algorithms commonly consisting of three sub phases:

1) Key generation symmetric or asymmetric algorithm.
2) Signing algorithm.
3) Signature verification algorithm

The symmetric key algorithm generates single key that is shared by sender and receiver. On other hand, the asymmetric key algorithm generates two keys: public and private keys. The public keys are shared between two parties; in contrast the private keys are keeping secret. During second phase signing algorithm the digital signature is generated by taken plain text i.e. private key, sensitive data, and message as input. After that, the sender sends the message along with generated signature to the intended recipient. Signature verification algorithm is executed at recipient end to ensure the received data [7]. A valid digital signature gives a receiver the reason to admit message and ensure the message was created and communicated by a known sender, not altered in transit. Digital signature has numerous schemes, such as RSA, DSA and ECDSA, which are used to impose the security of different transaction [7]. Digital signature schemes were enhanced in order to overcome some of vulnerabilities. Some improvement techniques of digital signature schemes are attained with respect to various perceptions.

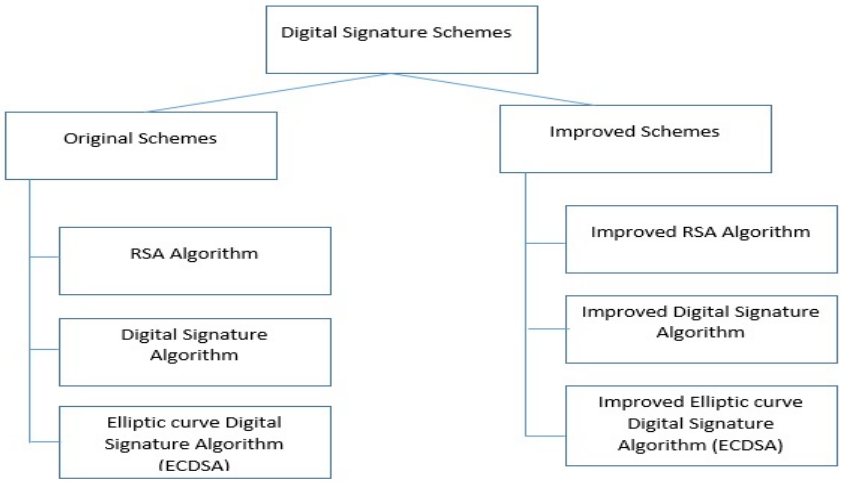

Figure 2. The improvement techniques of digital signature schemes that are attained with respect to various perceptions

In RSA, it is fault tolerance perspective, whereas in DSA, they are speed of operation computational perspective and longtime of computations perspective. And in ECDSA, they are efficiency perspective and speed of operation computational perspective

5. PERFORMANCE COMPARISON

The performance measurements have been categorized according to the dependent variables. References [8] regarding the chosen algorithms with respect to their performance and compared to the level of security provided.

Table 1. The performance measurements according to the dependent variables

	Algorithm Family	**Security level (in bits)**			
		80	**128**	**192**	**256**
RSA	Integer factorization	1024	3072	7680	15360
DSA	Discrete logarithm	1024	3072	7680	15360
ECDSA	Elliptic Curves	160	256	384	512

RSA and DSA algorithms are suspended from improvement of their performance for the reason that installing such algorithm on light-weight devices will adversely affect their performances and delay the decryption process. ECDSA in equivalent could be a auxiliary for RSA & DSA system, their comptability to be installed in any system with different memory sizes and CPU description and parameters, ECDSA provide the same level of security as RSA and DSA but with smaller keys: The lesser key sizes of ECDSA possibly allow for less computationally able light-weight devices and wireless systems to use cryptography for secure data transmissions, data verification and offers less heat generation and less power consumption, less storage space and offers an optimized memory and bandwidth and faster signature generation.

6. EXISTING SYSTEM

The scheme is apt for a signer who has limited computing capability like, a signer using his smart Card which stocks his secret key and displays a message on a external Key pair phase of this scheme is same as the ECDSA scheme.

6.1. Signature Generation

Using sender's private key, sender generates the signature for message M using the subsequent steps:
(1) Select a unique and unpredictable integer k in the interval [1,n-1]
(2) Compute $kg = (x_1,y_1)$, where x_1 is an integer
(3) Compute $r = x_1 \bmod n$; If r = 0, then go to step 1
(4) Compute h =H(M), where H is the SHA-512[9]
(5) Compute $s = k^{-1}(h + dr) \bmod n$; If s = 0, then go to step1
(6) The signature of sender for message M is the integer pair (r, s)

6.2. Signature Verification

The receiver can authorize the authenticity of sender's signature (r, s) for message M with the aid of execution the following:
(1) Obtain signatory A's public key (E, q, n, Q)
(2) Verify that values r and s are in the interval [1,n-1]
(3) Compute $w = s^{-1} \bmod n$.
(4) Compute h = H(M), where H is the same secure hash algorithm used by A.
(5) Compute u1 = hw mod n
(6) Compute u2 = rw mod n
(7) Compute u1G + u2Q = (x0,y0)
(8) Compute v = x0 mod n
(9) The signature for message M is verified only if v = r

6.3. A Possible attack

The secret key k used for signing two or more messages will have to be produced separately. In particular, additional secret k should be used for signing select messages, in any other case the private key d can also be recovered. However if a random or pseudo-random number generator is used, then the threat of making a repeated k value is negligible. If same secret k is used to produce signature of two different messages m_1 and m_2 then and there it will effect in two signatures (r,s_1) and (r, s_2).

- $s_1 = k^{-1}(h_1 + dr)$
- $s_2 = k^{-1}(h_2 + dr)$; where h1 = SHA512 (m_1) and h2 = SHA512 (m_2).
- $ks_1 - ks_2 = h1+dr-h2-dr$
- k = (h1-h2)/ (s1-s2)
- d = (ks-h)/r

7. PROPOSED SYSTEM

In this variant there is no need to find inverse in each key generation and signing section. This scheme is developed without modular inversion process in Signature generation and Verification algorithms.

7.1. Notations

To be appropriate in explanation of our work the elements are defined as
d: private key
Q: Public key

m: message
H() : a secure one-way hash function
r, s_1, s_2: Signature elements
q: field order
FR: field representation
a, b: coefficients
G: base point
n: Order of G
h: co-factor

7.2. Key pair Generation

Key pair d and Q made by the Signer as follows
INPUT: D= (q, FR, a, b, G, n, h)
(1) Choose a distinctive and unpredictable integer, d, within the interval [1, n-1]
(2) Compute Q ← (dg)
(3) Return (Q, d)
OUTPUT: Q, d

7.3. Signature Generation

The signer can sign message m as follows
INPUT: D= (q, FR, a, b, G, n, h), d, m
Begin
repeat
k = Random [1, 2,…, n-1]
P = kG
c=X-Co-ordinate (P)
e = H (m) mod n
s_1 = eck mod n
s_2 = (dc + 1)k mod n
R = eP
r = X-Co-ordinate(R)
until $r \neq 0$ and $s_1 \neq 0$ and $s_2 \neq 0$ return (r, s_1, s_2)
End
OUTPUT: Signature (r, s_1, s_2)

7.4. Signature Verification

To verify the signature (r, s_1, s_2) on message m, receiver does the following:
INPUT: D= (q, FR, a, b, G, n, h), Q, m, Signature (r, s_1, s_2)
Begin
ifr, s_1, s_2 doesn't belongs to [1,…, n-1] then
Return ("Reject the signature")
end if
e= H(m)
t = es_2
U1 = tG
U2 = s_1Q
W= U1 – U2
v= X-Co-ordinate (W)
if v = r then
Return ("Accept the signature")
else
Return ("Reject the signature")
end if
end
OUTPUT: Acceptance or rejection of the signature.

7.5. Proof Of Signature Verification

We begin with W=U1-U2
By substituting U1 with tG and U2 with s_1Q

$W = tG - s_1Q$

By substituting t with es_2 and Q with dG

$W = es_2G - s_1dG$

By substituting s_1 with ec and s_2 with (dc + 1)k

$W = e(dc +1)kG – ecdG$

$= edckG + ekG – eckdG$

$= ekG$

$= eP$

$= R$

v =X-Co-ordinate (W) and

r = X-Co-ordinate(R)

Therefore v = r.

K cannot be resolute although similar secret key is used to sign two different messages. So this System is not vulnerable to attack on same secret.

8. RESULTS AND DISCUSSION

ECC can be implemented in software and hardware [10]. Software ECC implementation provide moderate speed, higher power consumption and also have very limited physical security w.r.t key storage. Where as hardware implementation improves performance in terms of flexibility. Also hardware implementation provides greator security since they cannot be easily modified or read by an outside attacker. This section represents implementation results of our Proposed Scheme

Basepoint = (42582623172388835044654159270140906591363556877O,
20352011416290410787399145795734689202798264197O)
Jaya Bhaskar in genkey
Basepoint::genKey = (42582623172388835044654159270140906591363556877O,
20352011416290410787399145795734689202798264197O)
private_A = 340282366920938463463374607431768211455
EllipticCurve: y^2 = x^3 + 14615016373309029182036848327162830196537850593 24x +
16323579130616811054660491940327157953054834541 3 (mod
14615016373309029182036848327162830196537850593 27)
created successfully!
public_A = (19359627546068943863305713502614122336145146071 2,
85258563103004487371035250155333314837714566612 6)
Public_A on the curve is true

8.1. Signature Generation

Select random number = 14615016373309029182036871976068267798841648049 61

Compute base point * random number=P

P = (53115865784461915599516741479943270269709525770 5,
18034491864589497465121827332898921843168050957 6)

c = 5311586578446191559951674147994327026970952577 05

hex:-5d52e9cb5d889f6dd7ab9f28415d2c7bfd8659f3

dec:-53278516915716676152576641840073696483620620542 7

hash:[B@1c5fde0

Original Message: Paul hated school. He did not do his home

e = 92871646817373615667792077920608981504843728701 2

s_1=eck=73869772891774848176721914997527959239203013700096428838376461885758708494 402266060052176548136

s_2=(dc+1)k-->17042713598250844310553189773738841022750968520 4

R=eP = (98905772286823120676976338989980511065152918791 2,
20339143309476791259560039627836221859951852891 2)

r=x-co-ord(R) = 9890577228682312067697633898998051106515291879 12

8.2. Signature Verification

hex:-5d52e9cb5d889f6dd7ab9f28415d2c7bfd8659f3

dec:-53278516915716676152576641840073696483620620542 7

hash:[B@1b5340c

Original Message: Paul hated school. He did not do his home

E = 928716468173736156677920779206089815048437287012
t=es$_2$G = 660542466286145164871991432880014098834552069294
Compute U1=tG = (867656430810165309875458600806608047488460704882,
1293539316315317053662292883889929244439313430206)
ComputeU2=s$_1$Q = (71732824241819992935376226987838881939840564741 8,
1032047338562966998576808247653816832960909777 35)
ComputeW=U1-U2 = (98905772286823120676976338989980511065152918 7912,
2033914330947679125956003962783622185995185289 12)
v=x-coord(W) = 98905772286823120676976338989980511065152918 7912
We obtain v=r, hence Signature is accepted

We compare the results of ECDSA and our proposed system that presents the no of Point Addition and Scalar Multiplication operations for Signing and Signature Verification process. ECDSA uses inversion operation in both signing and Signature Verification but our Proposed System doesn't use any inversion operations in Signing and Signature Verification. We implemented original ECDSA and our proposed scheme and compared their performance over Elliptic Curve and presented the results below.

Table 2. Implemented original ECDSA and our proposed scheme and compared their performance over Elliptic Curve

Algorithm	ECDSA	Proposed Algorithm
No. of Secret keys	1	1
Inverse in Signing	Yes	No
No. of scalar Multiplication operations in signing	1	2
Inverse in Signature Verification	Yes	No
No. of Point Addition operations in Verification	1	1
No. of scalar Multiplication operations in Verification	2	2

From Figure 3 Proposed Signature scheme implemented poorly in signature generation since security is inversely proportional to performance of the system.

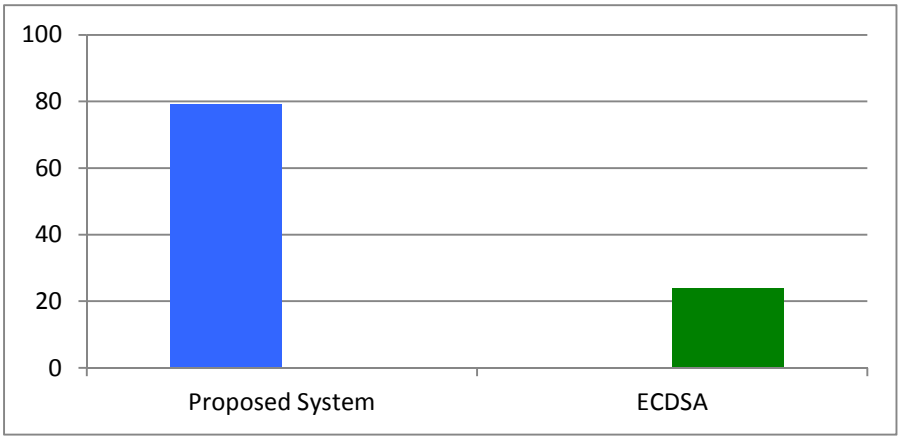

Figure 3. Proposed Signature scheme

From Figure 4, proposed scheme signature verification algorithm performed better when compared to the existing verification scheme. This is desirable because to the application-oriented point of view, message is authorized by the individual only once, but verification may be required many times.

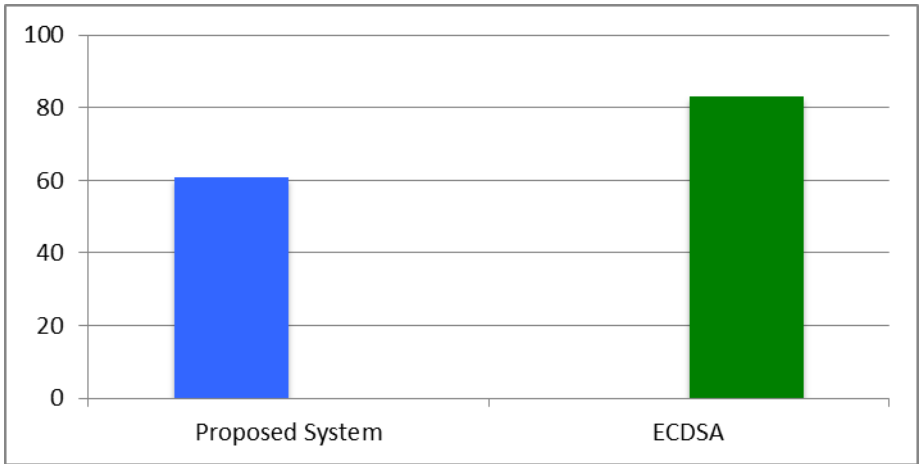

Figure 4. Proposed scheme signature verification algorithm

Our applications requiring Signature verification more frequently than Signature Generation, hence proposed scheme is best suitable for Internet of Things.

9. CONCLUSION

In the Existence system, if the same random number is generated which is used to sign the message, then there is a chance of decrypting the private key by the attacker. But in our proposed scheme, even if the same random number is used attacker can't decrypt the private key. Modular inversion operation is additional time consuming operation [11] for constrained devices. Our proposed Digital Signature scheme is developed without modular inversion process in Signature generation and Verification algorithms. But modular inversion operation is used in existence system. Considering the above, our proposed digital signature scheme is more secure and efficient when compared to the existing scheme.

REFERENCES

[1] V. S. Miller, "Use of Elliptic Curves in Cryptography," *Springer- Verlag Berlin Heidelberg*, 1986.
[2] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203-209, 1987.
[3] A. Khalique and K. S. S. Soodv, "Implementation of Elliptic Curve Digital Signature Algorithm," *International Journal of Computer Applications*, 2010.
[4] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203-209, 1987.
[5] V. Miller, "Uses of Elliptic Curve in Cryptography," *Advances in Cryptography, Proceedings of Crypto'85, Lectures notes on Computer Sciences, 218, Springer-Verlag*, pp. 417-426, 1986.
[6] D. Hankerson, *et.al*., "Guide to Elliptic Curve Cryptography".
[7] A. Roy and S. Karforma, "A Survey on Digital Signatures and Its Applications," *Journal of Computer and Information Technology*, vol. 3, pp. 45-69, 2012.
[8] T. Long, Xiaoxia L. I. U, "Two Improvements to Digital Signature Scheme Based on the Elliptic Curve Cryptosystem," *Proceedings of the 2009 International Workshop on Information Security and Application*, Nov-2009.
[9] V. Pallipamu, et al., "A Survey on Digital Signatures," *International Journal of Advanced Research in Computer and Communication Engineering* (*IJARCCE*), vol. 3, pp. 7243-724s 6, 2014.
[10] Marisa W. Po, *et al.*, "Issues in Elliptic Curve Crypyography implementation," *Internetworking Indonesial Journal*, vol/issue: 1(1), 2009.