Hybrid Signature Scheme Integrating Elliptic Curve Cryptosystem with Ong, Schnorr, and Shamir Digital Signature Scheme

Hisham A. Eltaib[*], **Roshdy AbdelRassoul,** ***SM IEEE*** **and Mohamed Saad Zaghloul**

Arab Academy for Science and Technology, Alexandria, Egypt

*hishamahmedeltaib92@gmail.com

Abstract. Transmitting data over unsecured Channels is one of the most indispensable issues which motivates researchers to study. In this paper, a new hybrid digital signature approach is going to be proposed which integrates the elliptic curve cryptosystem (ECC) into Ong, Schnorr, and Shamir signature (OSS). A self-invertible 8×8 key matrix will be utilized as a portion of the OSS digital signature. The strength of the proposed technique relies on combining couple of fundamental mathematical problems namely the discrete logarithm problem (DLP) and integer factorization problem (IFP). Elaborated results regarding security and time rendering analysis have been measured and confirmed its durability against diverse patterns of attacks. The submitted technique can be applied as a substitutional protocol if all the familiar systems are executed.

1. Introduction

Nowadays, Asymmetric key cryptosystem plays a significant role in securing any communication system. The existing cryptanalytic attacks stimulated researchers for calling a new digital signature approaches to overcome the widespread growth in security attacks. The importance of digital signature lays in providing privacy and data authentication. Consequently, a robust signature schemes have been clarified since the fabrication of the public key cryptography in 1970. Fifteen years later, the elliptic curve cryptosystem has been introduced by Koblitz and Miller [1, 2]. They contributed the cyclic group $((Z/pZ)^{*},.)$ more sophisticated than the ordinary multiplicative group where Z and p are set of all integers and prime number, respectively.

The ECC Construction has a considerable impact on public key cryptography [3], as a result, a new digital signature technique has been proposed by ElGamal. The foundation of this scheme based on the discrete logarithm problem. On the other hand, the traditional schemes are confronted with more and more complicated attacks. The strong security protocols previously represented and examined would be helpful, if the traditional techniques are completely expired [4].

This manuscript clarifies a new signature scheme which integrates the ECC into OSS signature scheme. The security of the proposed scheme relies on using a self-invertible 8×8 matrix as a portion of the OSS signature equation. Therefore, the diffusion and confusion complexity will be grown up. For the credibility of this work, the new approach has been examined and demonstrated its durability.

The upcoming sections are sequenced as follow: section 2 provides a brief introduction about elliptic curve cryptosystem and OSS signature scheme. Section 3 crystalizes the proposed Digital signature technique. Section 4 discusses an implementation example. Security strength is mentioned in section 5. Time rendering is measured in section 6. Finally, the manuscript is summarized in section 7.

2. Elliptic curve cryptosystem & OSS signature scheme

2.1. Elliptic curve cryptosystem

One of the most significant features which makes ECC an efficacious encryption algorithm is its capability to provide security requirements within a fewer mathematical operations (Multiplications and Additions), lower power consumption using small key length and less storage space. Consequently, it becomes an indispensable part of many applications [3] like mobile devices and network protocols [5, 6].

2.1.1. Definition.

The elliptic curve is a cubic curve defined by an algebraic equation in the form of:

$$E: y^2 \equiv x^3 + a.x + b \quad \bmod p \tag{1}$$

Where Z_p of $p > 3$ is a set of all points $(x, y) \in Z_p$

$$4.a^3 + 27.b^2 \neq 0 \quad \bmod p \ ,a, b \in Z_p \tag{2}$$

$E(Z_b)$ is defined as the elliptic curve (EC) group which contains all the points that fulfil the ECC equation and the point at the infinity O.

2.1.2. EC mathematical operations.

The basic mathematical operations of ECC functions will be declared in this section. Scalar Multiplication is the major fundamental operation on elliptic curve cryptosystem which consists of point summation and doubling [5, 6].

1. Condition 1: $P \neq Q$, the summation point (R=P+Q) can be estimated by drawing a line passing through P and Q, then the summation point (R) will be the mirror point of the third intersection point (R') as clarified in Figure 1(a).
2. Condition 2: $P = Q$, the summation point ($R = 2Q$) can be computed by drawing a tangent line through Q, then the doubling point (R) will be the mirror point of the second intersection point as illustrated in Figure 1 (b).

$R = (x_3, y_3)$

where

$$x_3 = \lambda^2 - x_1 - x_2 \quad \bmod p \tag{3}$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \quad \bmod p \tag{4}$$

and

$$\lambda = \frac{3{x_1}^2 + a}{2y_1} \quad \bmod p \qquad \text{if } P = Q \tag{5}$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \quad \bmod p \qquad \text{if } P \neq Q \tag{6}$$

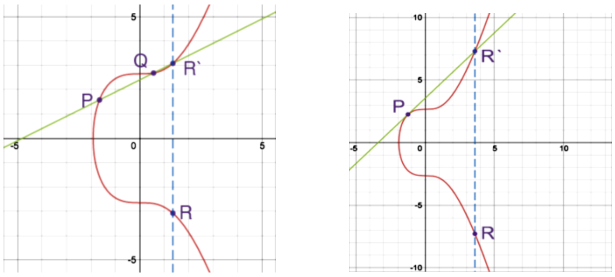

Figure 1. (a) Point summation [5]. (b) Point doubling [5].

2.2. OSS signature scheme

Fast and Secure are a primary feature in qualified signature scheme. If such algorithm is proposed, it will be applicable in many fields. A functional quadratic equation scheme has been designed by Ong, Schnorr, and Shamir. This scheme is characterized by speedy signature generation and verification as it requires only single inversion and two modular multiplications to generate the signature. However, three multiplications are essential for validation.

The durability of such scheme lays in hard solving of the quadratic equation $x^2 + ky^2 \equiv m \bmod n$, where m is the original text, k is the public key and n deduced from Euler's theorem [7]. The flowchart of OSS scheme is demonstrated as shown in Figure 2.

The receiver (verifier) will get the pair {x, y} to check if:

$$x^2 + ky^2 \overset{?}{=} m \quad \bmod n$$

Proof:

$$x^2 + ky^2 \equiv \left((r + mr^{-1}).2^{-1}\right)^2 + (-u^2).\left((r - mr^{-1}).2^{-1}.u^{-1}\right)^2$$
$$\equiv 2^{-2}[(r^2 + m^2r^{-2} + 2m) - (u^2.u^{-2})(r^2 + m^2r^{-2} - 2m)] \equiv 2^{-2}(2m + 2m) \equiv m \quad \bmod n$$

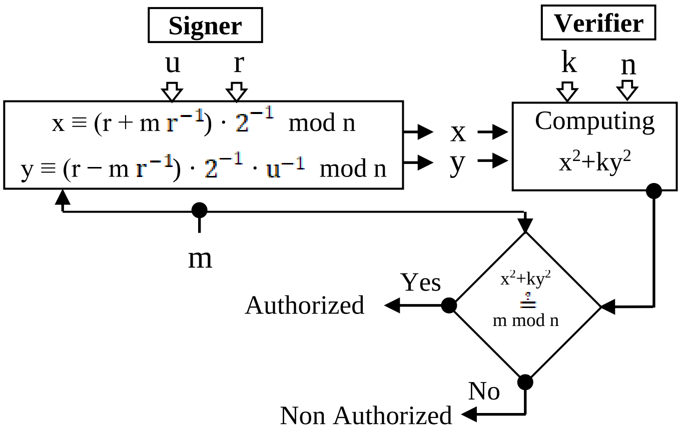

Figure 2. OSS digital signature scheme

3. ECC-OSS digital signature

A functional signature scheme will be stated in this section. It's clear that the classical OSS digital signature approved its deficiency against cryptanalytic attacks, as a result, the proposed scheme has been demonstrated. This modification has a significant effect on the system's durability which makes it more efficacious than the classical one. The expected time for signature generation will also be shrunk, as the key matrix and its inverse are the same.

The main parameters {G, p, a, b} of the elliptic curve digital signature algorithm (ECDSA) will be publicly shared between the two users terminals which are (User A) as the verifier and (User B) as the signer, where G is the base point, p = n = following prime number of [P×Q] and the EC function coefficients are {a, b}, then the private keys $\{n_A, n_B\}$ will be generated by each user terminal {A, B} respectively, which are a random integer numbers fluctuates between 1 and p-1.

The public key $\{P_A\}$ will be computed by the verifier as:

$$P_A = n_A.G \tag{7}$$

Thus, the signer calculates the initial key $\{K_I\}$, which results from the multiplication between the signer's private key $\{n_B\}$ and the public key $\{P_A\}$ as:

$$K_I = (x,y) = n_A.n_B.G = n_B.P_A \tag{8}$$

Then $$K_1 = (K_{11}, K_{12}) = x.G \qquad (9)$$
$$K_2 = (K_{21}, K_{22}) = y.G \qquad (10)$$

The signer will create the confidential key matrix (r). It's known that the inverse of any matrix doesn't coexist, if it's a non-invertible matrix. Therefore, the capability of generating the signer's signature will be unattainable. To overcome this difficulty, an 8×8 self- invertible confidential matrix will be computed $(r = r^{-1})$ so that, there is no need to generate the inverse confidential matrix by traditional technique as it will be created using the subsequent mechanism twice time in a row as follow [8, 9, 10]:

$$c = \begin{bmatrix} K_{11} & K_{12} & \vdots & K_{13} & K_{14} \\ K_{21} & K_{22} & \vdots & K_{23} & K_{24} \\ \cdots & \cdots & \vdots & \cdots & \cdots \\ K_{31} & K_{32} & \vdots & K_{33} & K_{34} \\ K_{41} & K_{42} & \vdots & K_{43} & K_{44} \end{bmatrix}$$

The proposed technique depends on partitioning the 4×4 auxiliary matrix (c) into four main sections as $c = \begin{bmatrix} K_A & K_B \\ K_C & K_D \end{bmatrix}$, where $K_A = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$, the three other sections will be computed as $K_B = I - K_A$, $K_C = I + K_A$, and $K_A + K_D = 0$, where (I) symbolizes the identity matrix.

This technique will be iterated twice time in a row to get the 8×8 self-invertible confidential matrix (r) from (c) as follows:

$$r = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & \vdots & K_{15} & K_{16} & K_{17} & K_{18} \\ K_{21} & K_{22} & K_{23} & K_{24} & \vdots & K_{25} & K_{26} & K_{27} & K_{28} \\ K_{31} & K_{32} & K_{33} & K_{34} & \vdots & K_{35} & K_{36} & K_{37} & K_{38} \\ K_{41} & K_{42} & K_{43} & K_{44} & \vdots & K_{45} & K_{46} & K_{47} & K_{48} \\ \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots \\ K_{51} & K_{52} & K_{53} & K_{54} & \vdots & K_{55} & K_{56} & K_{57} & K_{58} \\ K_{61} & K_{62} & K_{63} & K_{64} & \vdots & K_{65} & K_{66} & K_{67} & K_{68} \\ K_{71} & K_{72} & K_{73} & K_{74} & \vdots & K_{75} & K_{76} & K_{77} & K_{78} \\ K_{81} & K_{82} & K_{83} & K_{84} & \vdots & K_{85} & K_{86} & K_{87} & K_{88} \end{bmatrix}$$

After that, the signer's signature {x, y} had to be evaluated as:

$$x \equiv (r + mr^{-1}).2^{-1} \quad \mod n \qquad (11)$$
$$y \equiv (r - mr^{-1}).2^{-1}.u^{-1} \quad \mod n \qquad (12)$$

Where (m) represents the original data and the confidential matrix (r) which satisfies the condition $GCD(n, r) = 1$, the verifier receives the signature {x, y} and test whether

$$x^2 + ky^2 \overset{?}{=} m \quad \mod n \qquad (13)$$

The flowchart of the proposed algorithm is crystalized as shown in Figure 3.

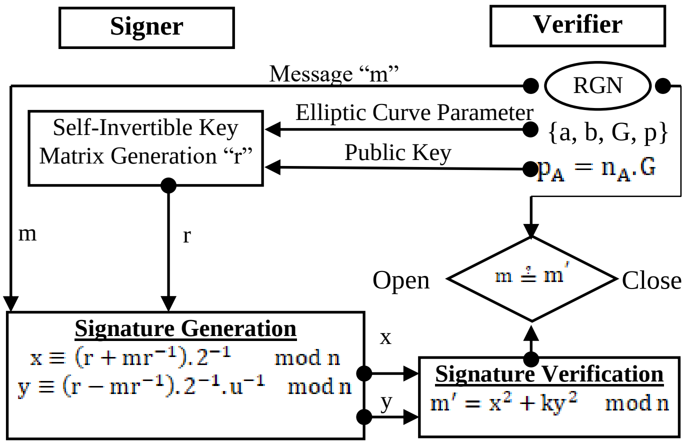

Figure 3. Flowchart of the ECC-OSS scheme

The following section emphasizes an example for ECC-OSS signature scheme.

4. ECC-OSS explanation example

Consider that the verifier matches the signer's signature and the EC function (E) is confirmed to be used between the two endpoint terminals with the following parameters:

$E: y^2 \equiv x^3 + 2x + 5 \quad \mod 37$

Where {a, b, P, Q} = {2, 5, 6, 6} respectively, so that p = n = following prime number of [6×6] = 37, which fulfill the equation:

$4a^3 + 27b^2 = (4.2^3) + (27.5^2) = 707 \quad \mod 37 \neq 0$

Table 1 comprehends 34 points which represents the order of $E_{37}(2,5) = 34$. The base point (G) is randomly selected from Table 1. So that, the EC function's parameters are {a, b, p, G} = {2, 5, 37, (16, 10)}.

Table 1. The Doubling Points of EC Function $E_{37}(2,5)$

$(3,1)$	$(3,36)$	$(4,15)$	$(4,22)$	$(6,14)$	$(6,23)$
$(9,7)$	$(9,30)$	$(10,10)$	$(10,27)$	$(11,10)$	$(11,27)$
$(16,10)$	$(16,27)$	$(18,8)$	$(18,29)$	$(20,4)$	$(20,33)$
$(21,13)$	$(21,24)$	$(22,2)$	$(22,35)$	$(26,13)$	$(26,24)$
$(27,13)$	$(27,24)$	$(31,6)$	$(31,31)$	$(33,9)$	$(33,28)$
$(34,3)$	$(34,34)$	$(35,17)$	$(35,20)$		

4.1. Keys computation process

1) The public keys will be generated by the verifier which will be transmitted to the signer to compute its keys according to the following steps:
   - Randomly selection of the verifier's private key as:

$$n_A = 18 \in [1,36]$$

   - Computing the public key as:

$P_A = n_A.G = 18(16,10) = (34,3)$

2) The confidential matrix (r) will be estimated using the received keys $\{n_A, P_A\}$ as:
   - Randomly selection of the signer's private key as:

$n_B = 26 \in [1,36]$

   - Computing K_I as:

$K_I = (x,y) = n_B.P_A = 26(34,3) = (16,27)$

   - Computing

$K_1 = x.G = 16(16,10) = (K_{11}, K_{12}) = (35,17)$

$K_2 = y.G = 27(16,10) = (K_{21}, K_{22}) = (16,27)$

   - Consider that $K_A = \begin{bmatrix} 35 & 17 \\ 16 & 27 \end{bmatrix}$, then the auxiliary matrix will be $c = \begin{bmatrix} 35 & 17 & 3 & 20 \\ 16 & 27 & 21 & 11 \\ 36 & 17 & 2 & 20 \\ 16 & 28 & 21 & 10 \end{bmatrix}$, and then the self-invertible confidential matrix (r) will be as:
   - Applying Euler's theory as:

$\Phi(PQ) = \Phi(36) = 12$

Randomly choice of the integer number (u) which satisfies the condition $GCD(u, \Phi(PQ)) = 1$, then the private key will be computed as:

$u^{-1} \quad \mod \Phi(n) = 11^{-1} \quad \mod 12 = 11$

   - Assuming the original data m=28, which will be used to get the signature as:

$x \equiv (r + mr^{-1}).2^{-1} \quad \mod n$

$$\equiv \begin{bmatrix} 8 & 6 & 25 & 31 & 25 & 31 & 12 & 6 \\ 10 & 3 & 27 & 30 & 27 & 30 & 10 & 7 \\ 4 & 6 & 29 & 31 & 33 & 31 & 4 & 6 \\ 10 & 36 & 27 & 34 & 27 & 1 & 10 & 36 \\ 4 & 6 & 25 & 31 & 29 & 31 & 12 & 6 \\ 10 & 36 & 27 & 30 & 27 & 34 & 10 & 7 \\ 4 & 6 & 25 & 31 & 33 & 31 & 8 & 6 \\ 10 & 36 & 27 & 30 & 27 & 1 & 10 & 3 \end{bmatrix}$$

$$y \equiv (r - mr^{-1}).2^{-1}.u^{-1} \mod n$$

$$\equiv \begin{bmatrix} 36 & 27 & 20 & 10 & 20 & 10 & 17 & 27 \\ 8 & 32 & 29 & 24 & 29 & 24 & 8 & 13 \\ 18 & 27 & 1 & 10 & 19 & 10 & 18 & 27 \\ 8 & 14 & 29 & 5 & 29 & 23 & 8 & 14 \\ 18 & 27 & 20 & 10 & 1 & 10 & 17 & 27 \\ 8 & 14 & 29 & 24 & 29 & 5 & 8 & 13 \\ 18 & 27 & 20 & 10 & 19 & 10 & 36 & 27 \\ 8 & 14 & 29 & 24 & 29 & 23 & 8 & 32 \end{bmatrix}$$

$$k = -u^2 \mod n = -11^2 \mod 39 = -10$$

- The public keys {n, K}, and signature {x, y} will be sent back to the verifier by the signer.

$$r = \begin{bmatrix} 35 & 17 & 3 & 20 & 3 & 20 & 34 & 17 \\ 16 & 27 & 21 & 11 & 21 & 11 & 16 & 26 \\ 36 & 17 & 2 & 20 & 1 & 20 & 36 & 17 \\ 16 & 28 & 21 & 10 & 21 & 9 & 16 & 28 \\ 36 & 17 & 3 & 20 & 2 & 20 & 34 & 17 \\ 16 & 28 & 21 & 11 & 21 & 10 & 16 & 26 \\ 36 & 17 & 3 & 20 & 1 & 20 & 35 & 17 \\ 16 & 28 & 21 & 11 & 21 & 9 & 16 & 27 \end{bmatrix}$$

4.2. Signature validation/verification

The verifier will test weather:

$$x^2 + ky^2 \overset{?}{=} m \mod n \equiv \begin{bmatrix} 28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 28 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 28 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 28 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 28 \end{bmatrix} \equiv m.I$$

5. Security standardization

A couple of strong attacks will be discussed in this section which will reflect the durability of the proposed scheme. Considering a third terminal is trying to estimate the signature [11, 12].

5.1. First cryptanalytic attack

Assuming a third terminal is attempting to fabricate the signature by estimating the second part {y}, assuming {x} is a fixed constant. A square root had to be computed to get the second part {y} which is as complex as factoring {n} and vice versa. Estimating {x} and fixing the second part {y} equals factoring {n}. Such quadratic equation is difficult to be solved.

Proof of correctness: Assume that algorithm (A) computes the private key using the possibly signatures of random message (m) and the public keys.

The factoring algorithm (B) has been designed depending on (A) as follow:

Step (1): Given $n = P \times Q$, where $\{P, Q\}$ are unknown.

Step (2): Choosing random integer number (u), then computing $u^{-1} \mod \Phi(n)$.

Step (3): Algorithm (A) requires signatures of random messages (m) to run, and then random messages (m) must be signed using the private key u^{-1}.

Step (4): Rendering algorithm (A) using the signatures and the public keys $\{K, n\}$.

Step (5): $u'$is computed by algorithm (A) as follow:

$$-u'^{\,2} \equiv k \mod \Phi(n) \tag{14}$$

Step (6): With a probability of $\frac{1}{2} u' = \pm u \mod \Phi(n)$, then the $GCD(\Phi(n), u' \pm u) > 1$, are the two prime numbers {P, Q}.
Step (7): According to the aforementioned steps, if $u' \neq \pm u \mod \Phi(n)$, then, choose another integer number (u) and redo the previous steps.
Step (8): After (t) rounds, the possibility of computing the factorization is $1 - 2^{-t}$.

5.2. Second cryptanalytic attack:

An efficacious solution to solve the quadratic equation $x^2 + ky^2 \equiv m \mod n$ has been represented by Pollard and Schnorr. The proposed technique approved that it's not essential to get the private key to make sure that the system is secured. The suggested scheme is designed to get the signature $\{x, y\}$ without having the private key (u^{-1}).
Step (1): This scheme based on the equality of

$$(x_1^{\,2} + ky_1^{\,2})(x_2^{\,2} + ky_2^{\,2}) = X^2 + kY^2 \qquad (15)$$

Where

$$X = x_1x_2 \pm ky_1y_2 \qquad (16)$$
$$Y = x_1y_2 \mp y_1x_2 \qquad (17)$$

Step (2): It's clear that equation (15) contains a couple of quadratic equations $(x^2 + ky^2)$. Defining the two variables $\{x', y'\}$ as follow:

$$x' = \frac{x}{y} \mod n \qquad (18)$$
$$y' = \frac{1}{y} \mod n \qquad (19)$$

Dividing the quadratic equation by y^2, then

$$\frac{x^2}{y^2} + k \equiv \frac{m}{y^2} \mod n \qquad (20)$$
$$x'^2 + k \equiv my'^2 \mod n \qquad (21)$$
$$x'^2 - my'^2 \equiv -k \mod n \qquad (22)$$

Step (3): Replacing (m) by (m') in the order of $O(\sqrt{k})$, then the solution of the quadratic equation using (m') gives the solution of original quadratic equation using (m).

The durability of ECC-OSS scheme implies that if another terminal has the main parameters of (EC) function $\{a, b, p, G\}$, and $\{P_A, k\}$ which are the public keys. Although, it will not be attainable to get the confidential matrix (r), as, it integrates (DLP) with (IFP). Similar scheme is really harsh to be broken. Performance comparison between RSA, ECDSA, and ECC-OSS digital signature is represented in Table 2.

Table 2. Performance comparison between RSA, ECDSA, and ECC-OSS digital signature

SCHEME	Security	Complexity	Domain	Key Creator	Execution Time	Verify/S	Sign/S
RSA	High	Integer Factorization Problem (IFP)	PC, Laptops, and Super Computers	Fast	Slow	Fast	Fast
ECDSA	High	Discrete Logarithm Problem (DLP)	Light-Weight Devices	Faster	Fast	Slow	Fast
ECC-OSS	High	(IFP)&(DLP)	Light-Weight Devices	Faster	Fast	Fast	Faster

6. Execution time measurement

Timing analysis plays a crucial role in measuring the efficiency of any digital signature scheme. The execution speeds of ECC-OSS signature generation and verification have been computed on a system with parameters as shown in Table 3.

Table 3. System's Parameters

Processor	Intel (R) Core ™ i5-2450M CPU@2.5 GHz, HD graphics 2.10 GHz
Installed Memory (RAM)	4.00 GB
System Type	Windows 7 Enterprise (64-bit operating system)
Executive Program	Wolfram Mathematica 8.0

The Execution speed of signature generation and verification are tabulated as shown in Table 4.

Table 4. Timing Analysis of Different Digital Signature Schemes (Signature Generation)

Rivest/Shamir/Adleman		**Elliptic curve DSA**		**(ECC-OSS) DSA**	
Key's Length	**Time (Second)**	**Key's Length**	**Time (Second)**	**Key's Length**	**Time (Second)**
1024	0.01	163	0.15	128	0.000932
2240	0.15	233	0.34	256	0.000992
7680	1.53	409	1.18	512	0.001004
15360	9.20	571	3.07	1024	0.001040

Nobody can deny that small key lengths will be more practical in limited storage space and Non-renewable power systems. The execution speed of the proposed scheme has been measured and compared to RSA and ECDSA [13, 14]. The resulted data represents the highly speed of ECC-OSS scheme with different key lengths which makes this proposal applicable in a lot of applications.

7. Conclusion

This paper proposes a new digital signature scheme; the strength of such scheme relies on integrating ECC with OSS as it combines between integer factorization problem and discrete logarithm problem. The resulted data in terms of implementation, security standardization, and timing analysis approved its durability and efficiency against cryptanalytic attacks. The ECC-OSS scheme works with a small key length which gives it the priority to work in low power consumption and minimum storage space systems. Therefore, this scheme is viable in a lot of applications such as IOT, RFID, and IFF.

References

[1] Darrel Hankerson, Alfred Menezes, and Scott Vanstone, "Guide to Elliptic Curve Cryptography," *Springer-Verlag Professional Computing Series*, pp. 11-12, 2004.
[2] Christof Paar, "Applied Cryptography and Data Security", version 2.5, Lecture Notes, Ruhr-Universit at Bochum, Germany, Jan. 2005.
[3] V. Miller, "Uses of Elliptic Curves in Cryptography," *Proceedings of Crypto'85, Lecture Notes in Computer Science*, v. 218, pp. 417-426, 1985.
[4] Lawrence C. Washington, "Elliptic Curves Number Theory and Cryptography,"*Taylor & Francis Group, Second Edition*, 2008.
[5] N. Koblitz, "Elliptic Curve Cryptosystem," *Mathematics of Comp. 48*, pp. 203-209, 1987.
[6] Lo'ai Tawalbeh, Moad Mowafi, and Walid Aljoby, "Use of Elliptic Curve Cryptography for Multimedia Encryption," *IET Information Security*, vol. 7, no 2, pp. 67–74, 2012.

[7] Ong, Schnorr, and Shamir, "An Efficient Signature Scheme Based on Quadratic Equations," *proceedings of the 16'th symposium on theory of computing*, pp. 208–216, 1984.

[8] Ounasser Abid, Jaouad Ettanfouhi, and Omar Khadir, "New Digital Signature Protocol Based on Elliptic Curve," *International Journal on Cryptography and Information Security (IJCIS),* vol.2, no.4, pp. 13-19, 12 Jan. 2013.

[9] Ziad E. Dawahdeh, Shahrul N. Yaakob, and Rozmie Razif bin Othman, "A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher," *Journal of King Saud University - Computer and Information Sciences*, V. 30, # 3, July 2018, pp. 349-355.

[10] Laiphrakpam Dolendro Singh and Khumanthem Manglem Singh, "Implementation of Text Encryption using Elliptic Curve Cryptography," *Eleventh International Multi-Conference on Information Processing*, vol. 54, pp. 73-82, 2015.

[11] Pollard and Schnorr, "An efficient solution of the congruence x^2 + ky^2 = mpmod{n}," *IEEE Transactions on Information Theory*, V. 33, # 5, pp. 702-709, 1987.

[12] W. Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, Fourth Edition, Nov. 2005.

[13] Hassan M. Elkamchouchi, Ali E. Takieldeen, and Mahmoud A. Shawky, "An Advanced Hybrid Technique for Digital Signature Scheme", *5th International Conference on Electrical and Electronics Engineering (ICEEE 2018)*, 5 May 2018.

[14] Nicholas Jansma and Brandon Arrendondo," Performance Comparison of Elliptic Curve and RSA Digital Signatures," Tech. Rep. MI, University of Michigan, Ann Arbor (May 2004).