Efficient and Secure ECDSA Algorithm and its Applications: A Survey

Mishall Al-Zubaidie[1,2], Zhongwei Zhang[2] and Ji Zhang[2]

[1]Thi-Qar University, Nasiriya 64001, Iraq
[2] Faculty of Health, Engineering and Sciences,
University of Southern Queensland, Toowoomba, QLD 4350, Australia

Abstract: Public-key cryptography algorithms, especially elliptic curve cryptography (ECC) and elliptic curve digital signature algorithm (ECDSA) have been attracting attention from many researchers in different institutions because these algorithms provide security and high performance when being used in many areas such as electronic-healthcare, electronic-banking, electronic-commerce, electronic-vehicular, and electronic-governance. These algorithms heighten security against various attacks and the same time improve performance to obtain efficiencies (time, memory, reduced computation complexity, and energy saving) in an environment of constrained source and large systems. This paper presents detailed and a comprehensive survey of an update of the ECDSA algorithm in terms of performance, security, and applications.

Keywords: ECDSA, coordinate system, fault attack, scalar multiplication, security.

1. Introduction

Public key encryption algorithms such as elliptic curve cryptography (ECC) and elliptic curve digital signature algorithm (ECDSA) have been used extensively in many applications [1] especially in constrained-resource environments due to the effectiveness of their use it in these environments. These algorithms have appropriate efficiency and security for these environments. Constrained resource environments such as wireless sensor network (WSN), radio frequency identifier (RFID), and smart card require high-speed, low consumption ability, and less bandwidth. ECC has been considered to be appropriate for these constrained-source environments [2]. These algorithms provide important security properties. For example, ECDSA provides integrity, authentication, and non-repudiation. ECDSA has been proven to be efficient in its performance because it uses small keys; thus the cost of computation is small compared with other public key cryptography algorithms, such as Rivest Shamir Adleman (RSA), traditional digital signature algorithm (DSA) and ElGamal. For example, ECDSA with a 256-bit key offers the same level of security for the RSA algorithm with a 3072-bit key [3, 4]. Table 1 [5, 6, 7] shows a comparison of key sizes for public key signature algorithms.

The preservation of efficiency and security in the ECDSA is important. On one hand, several approaches have been developed to improve the efficiency of the ECDSA algorithm to reduce the cost of computation, energy, memory, and consumption of processor capabilities. The operation that consumes more time in ECC/ECDSA is the point multiplication (PM) or scalar multiplication (SM). ECC is used PM for encryption and decryption, while ECDSA is used this operation to generate and verify the signature [8]. One can improve the PM efficiency by improving finite field arithmetic (such as inversion, multiplication, and squaring), elliptic curve model (such as Hessian and Weierstrass), point representation (such

as Projective and Jacobian), and the methods of PM (such as Comb and Window method) [9]. Many researchers have made improvements to the PM to increase the performance of the ECC/ECDSA as we will see in Section 4.

On the other hand, the security improvement in ECDSA is no less important than its efficiency because this algorithm is designed primarily for the application of security properties. ECDSA, like previous algorithms, may possibly suffer from some of the security vulnerabilities such as random weak, bad random source [1] or leaking bits of the private key. Also, many researchers have made improvements to close security gaps in the ECC/ECDSA algorithm by providing countermeasures against various attacks. But when selecting countermeasure there should be a balance between security and efficiency [10]. To maintain the security of these algorithms it is important to use finite fields (either prime or binary) recommended by credible institutions such as the Federal Information Processing Standard (FIPS) or National Institute of Standards and Technology (NIST). The choice of appropriate curves and finite fields according to the authoritative organizations' standards leads to secure ECDSA's implementations [11]. Therefore, we note from the above that any encryption algorithm or signature should possess a high performance and security level.

1.1 Our Contributions

Our contribution in this survey is to provide an updated study of three important parts in ECC/ECDSA. These items can be summarized as follows:

- Presenting a study on the efficiency of ECC/ECDSA in terms of speed, time, memory, and energy.
- Investigating the security problems of the ECC/ECDSA and countermeasures.
- Giving a description of the most important applications that use ECC/ECDSA algorithms.

1.2 Structure of the paper

The remainder of this paper is structured as follows: Section 2 provides basic concepts and general information about ECC and ECDSA algorithms. Section 3 describes existing surveys about ECC/ECDSA. Efficiency improvement on ECDSA algorithm is presented in Section 4. In Section 5, we will show the security improvement on ECDSA algorithm through using countermeasures against attacks. The ECDSA applications are described in Section 6. Finally, we will present the conclusion and future work on this survey in Section 7.

2. Preliminaries of ECDSA

In this section, we will present fundamentals about elliptic curve cryptography (ECC) and basic concepts of the ECDSA algorithms.

Table 1. Keys sizes and some information for public key algorithms

Algorithm	Keys sizes					Ratio	Author(s)	Year	Mathmatical problem	Other algorithms
RSA							Rivest, Shamirand, and Adleman	1978	Integer factorization	Rabin
Elgamal	1024	2048	3072	7680	15360		Taher Elgamal	1985	Multiplicative group	Schnorr, Nyberg-Rueppel
DSA						1:6-30	David W. Kravitz	1991		
ECDSA	160-223	224-255	256-383	384-511	512-more		Scott Vanstone	1992	Elliptic curve discrete log. (ECDLP)	ECIES

2.1 ECC

ECC has been used to encrypt data to provide confidentiality in the communications network with limited capacity in terms of power and processing. This algorithm was independently proposed by Neal Koblitz and Victor Miller in 1985 [12]. It depends on the discrete logarithm problem (DLP) which is impervious to different attacks when selecting parameters accurately [13], i.e difficulty obtaining k from P and Q (where k is integer and, P and Q are two points on the curve). Small parameters used in ECC help to perform computations quickly. These computations are important in constrained-source environments that require processing power, memory, bandwidth or power consumption [7]. ECC provides encryption, signature and keys exchange approaches [12]. Many operations are performed in ECC algorithms (shown in four layers) as shown in Figure 1 [14].

ECC uses two finite fields (prime field and binary field). The binary field uses two types to represent basis (normal and polynomial basis) [7], and is well suited to implementation in hardware [14]. Let F_q indicates field type, if $q=p$ (where $p > 3$) then ECC uses prime field (F_p). In the second case, if $q=2$ then ECC uses binary field (F_{2^m}) where m is the prime integer [15]. ECC consists of a set of points (x_i, y_i), where x_i, y_i are integers and the point at infinity (O) provides an identity for Abelian group rule that satisfies long form for Weierstrass equation (with Affine coordinate).

$$y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6 \qquad (1)$$

When prime field is used over ECC, the simplified equation is as follows:

$$y^2 = x^3 + ax + b \qquad (2)$$

Where $a, b \in F_p, 4a^3 + 27b^2 \neq 0 (mod p)$. The law of chord-and-tangent is used in ECC to add two points on the curve. Let us suppose that P and Q are two points on the curve; these two points have coordinates $(x_1, y_1), (x_2, y_2)$ respectively and the sum of these two points is equal to a new point $R(x_3, y_3)$ (i.e $P(x_1, y_1)+Q(x_2, y_2)=R(x_3, y_3)$) [7]. ECC uses two operations for addition that are point addition($P+Q$) and point doubling ($P+P$) (Figure 1), as in the following equations:
In the case of the point addition $(P + Q)$ where P and $Q \in E(F_p)$:
with using the slope $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$

$$x_3 = \lambda^2 - x_1 - x_2, \ y_3 = \lambda^2(x_1 - x_3) - y_1 \qquad (3)$$

In the case of the point doubling $(P + P)$ where $P \in E(F_p)$:
with using slope $\lambda = \frac{3x_1^2 + a}{2y_1}$

$$x_3 = \lambda^2 - 2x_1, \ y_3 = \lambda^2(x_1 - x_3) - y_1 \qquad (4)$$

When the binary field is used over ECC, the simplified equation is as follows:

$$y^2 + xy = x^3 + ax^2 + b \qquad (5)$$

Algorithm 1 ECC encryption and decryption algorithm:

1: Alice and Bob use same parameters domain $D = \{a, b, q, G, n, h\}$, where a, b are coefficient, q is field type, G is base point, n is order point and h is cofactor.
2: Alice selects random integer d_A as private key.
3: Alice generates public key $Q = d_A G$ and sends Q and G to Bob.
4: Bob receives message m from Alice, selects random integer as private key d_B where $d_B \leq n$.
5: Bob encrypts message m with point P in elliptic curve $E(F_q)$.
6: Bob computes $C_1 = P + d_B Q, C_2 = d_B G$ and sends C_1 and C_2 to Alice.
7: Alice receives Bob's message and decrypts the message by computing $C_1 - d_A C_2$ to obtain plaintext.

Where $a, b \in F_{2^m}, b \neq 0$ [7], as addition operations are in the following form:
In the case point addition $(P+Q)$ where P and $Q \in E(F_{2^m})$:
with using the slope $\lambda = \frac{y_1 + y_2}{x_1 + x_2}$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a, \ y_3 = \lambda(x_1 + x_3) + x_3 + y_1 \quad (6)$$

In the case point doubling $(P + P)$ where $P \in E(F_{2^m})$:

$$x_3 = x_1^2 + \frac{b}{x_1^2}, \ y_3 = x_1^2 + (x_1 + \frac{y_1}{x_1})x_3 + x_3 \qquad (7)$$

ECC operations for encryption and decryption are explained through the algorithm 1 [16].

2.2 ECDSA

ECDSA algorithm is used to warrant data integrity to prevent tampering with the data. This algorithm was proposed by Scott Vanstone in 1992. Data integrity of the message is important in the networks because the attacker can modify the message when it is transferred from source to destination [17]. Many organizations use it as standard such as ISO (1998), ANSI (1999) and, IEEE and NIST (2000) [7]. This algorithm is similar to the digital signature algorithm (DSA), where both algorithms depend on the discrete logarithm problem (DPL), but ECDSA algorithm uses a set of points on the curve and the generating keys are small. ECDSA algorithm with key length 160-bit provides the equivalent for symmetric cryptography with a key length of 80-bit [17]. It is dramatically convenient for devices with constrained-source because it uses small keys and provides computation speed in the signature [18]. Moreover, four point multiplication operations used in ECDSA algorithm: one is in public key generation, one for signature generation and two for signature verification [19]. In addition, this algorithm consists of three procedures: key generation, signature generation, and signature verification. These operations are explained as follows:
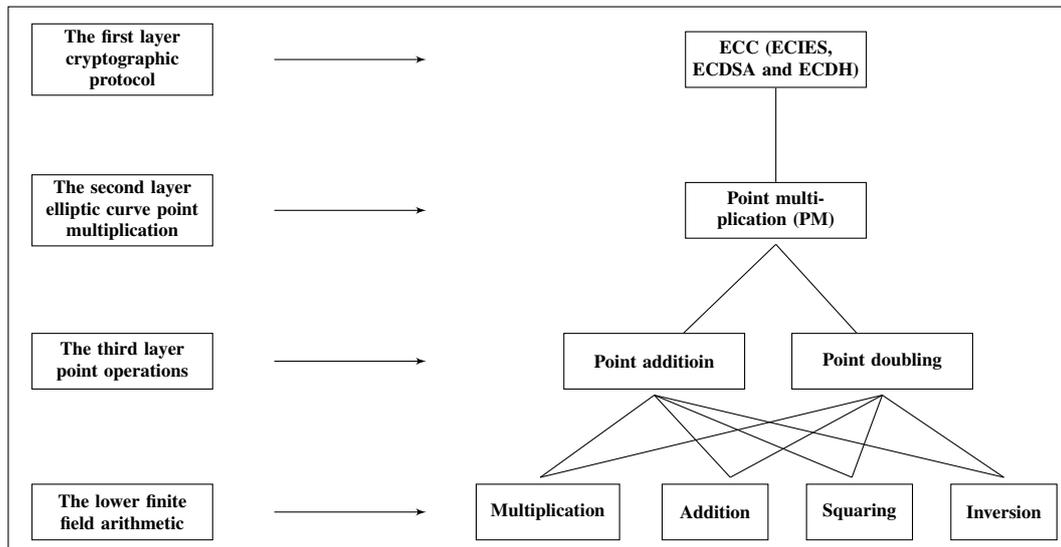
Figure 1. Arithmetic operations in ECC hierarchy

- **Key generation:**
  1 Select a random or pseudorandom integer d in the interval $[1, n-1]$.
  2 Compute $Q = dG$
  3 Public key is Q, private key is d.
- **Signature generation:**
  1 Select a random or pseudorandom integer k, $1 \leq k \leq 1$.
  2 Compute $kG = (x_1, y_1)$ and convert x_1 to an integer $\bar{x}_1$.
  3 Compute $r = x_1 \bmod n$. If $r = 0$ then go to step 1.
  4 Compute $k^{-1} \bmod n$.
  5 Compute SHA-1(m) and convert this bit string to an integer e.
  6 Compute $s = k^{-1}(e + dr) \bmod n$. If $s = 0$ then go to step 1.
  7 Signature for the message m is (r, s).
- **Signature verification:**
  1 Verify that r and s are integers in the interval $[1, n-1]$.
  2 Compute SHA-1(m) and convert this bit string to an integer e.
  3 Compute $w = s^{-1} \bmod n$.
  4 Compute $u_1 = ew \bmod n$ and $u_2 = rw \bmod n$.
  5 Compute $X = u_1 G + u_2 Q$.
  6 If $X = \theta$ then reject the signature. Otherwise, convert the x-coordinate x_1 of X to an integer $\bar{x}_1$, and compute $v = \bar{x}_1 \bmod n$.
  7 Accept the signature if and only if $v = r$.

ECDSA algorithm becomes unsuitable for signing messages (integration) if used poorly and incorrectly. Validation of domain parameters is important to ensure strong security against different attacks. This algorithm becomes strong if the parameters are well validated [20, 21]. The authors' recommendations are to update the validation scheme.

2.3 ECDSA Implementations on Constrained Environments

The performance and security presented by ECDSA algorithm, make it suitable for use in several implementations on WSN, RFID, and smart card. Digital signatures in ECDSA have better efficiency in devices with constrained-resource than DSA and RSA. Many authors have pointed to the possibility of using ECDSA with environments constrained-resource (memory, energy, and CPU capability). In the following Subsections, we will explain using ECDSA with these implementations.

2.3.1 Wireless Sensor Network (WSN)

A WSN consists of a group of nodes that communicate wirelessly with each other to gather information about a particular environment in various applications [22]. This network often has restricted sources such as energy and memory. Therefore, a WSN needs efficient algorithms to reduce the complexity of the computation in order to increase the length of the network lifetime. In addition, it requires a high level of security to prevent attacks. Several researchers have investigated the use of ECDSA algorithm in WSN and explained that ECDSA is convenient for WSN.

For example, Wander et al. [23] presented a study in energy for the public key cryptography (ECC/ECDSA, RSA) on sensor node Mica2dot with Atmel ATmegal 128L (8bit). They found that transmission cost is double the receiving cost. They analysed signatures in ECDSA with a key of bit-160 and RSA with a key of bit-1024, signature verification cost in ECDSA is larger than a signature generation while RSA verification is smaller than a signature generation. They noted that ECDSA has less energy cost than RSA. They concluded that ECC/ECDSA is more effective than RSA and feasible in constrained-source devices (WSN) because it generates small keys and certificates with same security level in RSA. Also, implementation was applied for 160-bit ECC/ECDSA and 1024-bit RSA algorithms on sensor node MICAz [24]. The authors used hybrid multiplication to reduce access memory. ECDSA results on MICAz are signature generation=1.3s and signature verification = 2.8s. For the purpose of comparison, the authors implemented ECDSA also on TelosB, where MICAz results were slightly less than TelosB's results. The authors proved the possibility of using RSA and ECDSA on WSN. In addition, ECDSA (SHA-1) and RSA (AES) were analysed in several types of sensor nodes (Mica2dot, Mica2, Micaz, and TelosB) in terms of energy and time [25]. ECDSA uses short keys (160-bit), which reduces memory, computation, energy and data size transmitted and thus is better than the RSA. De Meulenaer et al. [26] discussed the cost evaluation of energy (communication, computation) on the

WSN (TelosB and MICAz) through the Kerberos key distribution protocol (symmetric encryption) and ECDH-ECDSA key agreement protocol (asymmetric encryption). They noted that TelosB sensor node consumes less power than MICAz and Kerberos performs better than ECDH-ECDSA. Moreover, Fan and Gong [27] implemented ECDSA on Micaz motes with the binary field (163-bit). They improved signature verification via cooperation idea of the adjacent nodes. Also, ECDSA's implementation was presented in sensor node (IRIS) [28], but because this node supports 8-bit of the microcontroller the author modified the SHA-1 code from 32 bits original to 8 bits. Through implementation, the original algorithm is better in size and time than a modified algorithm. The author explained the possibility of using ECDSA algorithm with the sensor node (IRIS) held 8-bit microcontroller.

Recently, researchers in [29, 30] have applied the ECDSA algorithm as a light-weight authentication scheme in the WSN. This demonstrates the effectiveness and efficiency of using ECDSA in WSN in terms of security and performance.

2.3.2 Radio Frequency Identifier (RFID)

Another implementation of ECDSA is Radio Frequency Identifier (RFID). RFID is one of the technologies used in wireless communication. This technology has been used significantly in various fields, but it suffers from constrained-resources such as area and power. Therefore, many researchers considered ECDSA algorithm to be the best choice in RFID tags because ECDSA is not required to store private/public keys as in symmetric cryptography algorithms.

The hardware's implementation of ECDSA algorithm has been widely used on RFID technology [31]. Authors have used ECDSA algorithm to authenticate the entity. They have applied this algorithm to the prime field (F_{p160}) according to SECG standards. They accelerated multiplication algorithm by combining integer multiplication and fast reduction. Through implementation, they get good results in reducing chip area and power consumption ($860\mu w$ in 1MHz). They concluded that ECDSA computation requires 511864 clock cycles. Also, Hutter et al. [32] proposed ECDSA processor with RFID tag. This processor has security services (authentication, integrity, and non-repudiation). Authors used this processor to authenticate between tag reader and RFID tag. They used several countermeasures to prevent side channel attacks such as Montgomery ladder algorithm, randomized Projective coordinate and k randomness through SHA-1. Through results, the computation cost of the signature in this processor is 859188 clock cycles (127ms at 6.78 MHz) and the area is 19115 GE. Implementation of ECDSA algorithm is presented with prime field (F_{p160}) on RFID [33]. Authors exchanged SHA-1 hash algorithm with KECCAK hash to reduce running time in ECDSA on RFID. Also, they used a fixed-base comb with w=4 to accelerate and reduce hardware requirements. Results show that their scheme gets an area of 12448 GEs, power consumption 42.42 μw in 1MHz and signature generation is less than 140 kcycles. With these results, their scheme competes with other schemes for the binary and prime fields.

Recently, the ECDSA with password-based encryption in [34] has been adopted to improve security and privacy in RFID. Authors pointed out that lightweight processes performed by ECDSA in data signing are significantly effective in RFID.

Also, Ibrahim and Dalkılıç [35] used ECDSA with the Shamir scheme to secure RFID technology. They applied the Shamir scheme to reduce the cost of two scalar multiplications to one. Finally, ECDSA has been implemented to secure RFID in IoT applications in [36] have implemented. Authors proposed a shopping system, during the analysis and evaluation, arguing that ECDSA is suitable to sign users requests in the shopping system.

2.3.3 Smart Cards

A smart card is a novel way for authentication as it contains important information for users. This technology is used by several algorithms to implement authentication mechanisms, such as RSA, DSA and ECDSA. ECDSA algorithm is used in this area largely because of the advantages found in ECDSA. The design and implementation of ECC/ECDSA algorithms have been investigated and they are used in constrained-source devices like smart cards [12]. The authors used a java card that supports the java language and the environment used is next generation integrated circuit card (NGICC). Results showed that the ECC/ECDSA is better than RSA. The results also pointed to the possibility of using these algorithms in the other wireless devices constrained-source. Moreover, the EC algorithms (ECDSA and ECDH) with pairing were used on a Java card [37]. ECC requires little data to move between the card reader and the card, compared with the RSA. Furthermore,Savari and Montazerolzohour [38] concluded that the ECC/ECDSA algorithm is more efficient in terms of power, storage and speed than RSA on the smart card. On the other hand, Bos et al. [1] discussed using ECDSA with Austrian e-ID (smart card). They noted that ECDSA uses the same keys many times; therefore, they pointed to improved randomness in ECDSA. In 2017, Dubeuf et al. [39] discussed the security risks of the ECDSA algorithm when applied to a smart card. They proposed the development of a scalar operation algorithm when applying the Montgomery ladder method in ECDSA. They described that ECDSA offers a security solution for their smart card implementations.

3. Existing Surveys

In this section, we will present the existing surveys in ECC/ECDSA. In this survey, we will focus on a study to many of the aspects of ECC/ECDSA algorithm. To start with beginning, we will present these articles and then explain the difference between our research and existing studies. Table 2 lists existing surveys for the ECC/ECDSA algorithm.

- **Performance and efficiency**
  First of all, in [40] performance and flexibility have been investigated in the ECC algorithm with accelerators through hardware implementations. Many points in hardware implementations for ECC were discussed such as selecting curves, group law, PM algorithms, and selection of coordinates. In addition, it was pointed out that the architecture of multiple scalar multiplication in ECDSA's verification should supported because this architecture leads to efficiency in hardware's implementations. Much research has pointed out that using hardware's accelerators leads to high performance, but it sacrifices flexibility, where reduction circuits should be used to retrieve the flexibility feature. Similarly, Driessen et al. [17] compared many different signature schemes

Table 2. Different between existing surveys and our research

Paper	Contribution of existing surveys	Aspect	Year
[40]	Efficiency and flexibility in hardware implementations	Efficiency	2007
[17]	Compared many different signature schemes		2008
[41]	Attacks and countermeasures	Security	2010
[42]	Hard problems in public cryptography algorithms		2011
[43, 10]	SCA and fault attacks, and countermeasures		2012,2013
[44]	Security techniques in WSN	Applications	2015
[45]	Attack strategies in Bitcoin and Ethereum		2016
[6]	ECC/ECDSA with some applications		2017
This work	Classification of efficiency, security methods and applications with updated contributions	Efficiency Security Applications	

(ECDSA, XTR-DSA, and NTRUSing) in terms of energy consumption, memory, keys length and signature, and performance. Through implementation, the authors found that NTRUSing algorithm is the best in term performance and memory. However, NTRUSing algorithm suffer from security weakness against attacks.

- **Security and countermeasures**
A detailed study in [41] on attacks and countermeasures in ECC algorithm is presented. The authors divided attacks to passive and active attacks. They explained that the countermeasure for a specific attack may be vulnerable to other attacks, whereas countermeasures should have been selected carefully. Therefore, the authors have made some recommendations in selecting countermeasures. Some surveys have studied public cryptography algorithms in terms of computation of hard problems (integer factorization problem(IFP), discrete logarithm problem (DLP), lattices and error correcting codes) in quantum and classical computers [42]. The authors described RSA, Rabin, ECC, ECDH, ECDSA, ElGamal, lattices (NTRU) and error-correcting code (McEliece cryptography), as they pointed out that ECC provides a higher security level than other cryptosystems; in addition, it presents advantages such as high speed, less storage, and smaller keys sizes. But they did not discuss the use of ECC/ECDSA in applications and implementations of different technologies. Meanwhile, the authors in [43, 10] explained physical attacks on ECC algorithms, where they focused on two known physical attacks: side channel analysis (SCA), and fault attacks. They also described many attacks including these two types, as they presented countermeasures against these attacks such as simple power analysis (SPA), differential power analysis (DPA) and fault attack (FA) countermeasures. Also, some recommendations were presented for countermeasures that add randomness, countermeasures selection, and implementation issues. However, none of these papers investigated non-physical attacks on public key signature algorithms such as ECDSA.

- **Implementation and applications**
A study on security techniques has investigated wireless sensor networks (WSNs) [44]. It focused on three features in WSN security: key management, authentication, and secure routing. This study pointed out that ECC algorithm was convenient for constrained-resource devices. In addition, a survey on attack strategies was given in relation to ECC and ECDSA algorithms in Bitcoin and Ethereum [45]. The author pointed out that different standards for curves (such as ANSI X9.63, IEEE P1363, and safecurves), where this survey focused on safecurves with SECP256k1 through using ECDSA, as this paper

referred to safecurves as one of the strongest curves standards. The author suggested many basic points to prevent attacks on ECDSA or ECC. Finally, Harkanson and Kim [6] compared RSA and ECC/ECDSA, and pointed out that ECC/ECDSA exhibited the highest performance with the same level of security from RSA. They noted that 69% of websites use ECC/ECDSA, 3% used RSA and the rest used other algorithms. They also described ECC with some applications (such as vehicular communication, e-health and iris pattern recognition). However, they had a duplication between implementation and application. For example, RFID is a technology that can be used to implement a particular application.

In our survey, we study the ECDSA algorithm differently to previous studies. First, we integrate three aspects (efficiency, security, and applications) into one search. Second, systematically, we provide different details (ECDSA aspects) of previous studies. Finally, we provide an updated explanation of all these aspects in ECDSA.

4. Efficiency Improvement on ECDSA

In this section, we will discuss the improvement of ECDSA's efficiency in many ways such as scalar multiplication, coordinate system, and arithmetic operations. In each subsection, successive improvements to several authors will be explained.

4.1 Efficiency Improvement of Scalar Multiplication

This section includes many strategies to improve scalar multiplication; these are efficient in terms of scalar representation, curve operations, and ECDSA arithmetic operations.

4.1.1 Representation Improvement of the Scalar

In this section, we explain the scalar representation methods on scalar multiplication. Implementation of scalar multiplication take a large amount of time [46, 19] in ECDSA and ECC algorithms. Scalar multiplication takes more than 80% for running time in operation of key computation in sensor devices [13, 47]. Scalar multiplication (SM) or point multiplication (PM) is a set of point addition $(P+Q)$ and point doubling $(P + P)$ that generates kP (where k is a large integer and P is a point on the curve) through $P + P...(k$-times)[48, 49]. Representation of 25 points in traditional double-and-add (D&A) method presents as the following [50].

$25P = 2(2(2(P+2P)))+P$

Or

$25P = 2(2(2(2P)+P))+(2(2P)+P)$

The efficient and fast implementation of ECC algorithm and its derivatives are needed to accelerate SM implementation [51]. As we can see from step 5, signature verification in ECDSA algorithm requires 2 scalar multiplications ($u_1G + u_2Q$) [52] that require operation of a complexity computation.

SM uses three operations: inversion, squaring and multiplication (where it takes the notation i, s and m respectively) that is considered expensive for an ECC algorithm [48]. These operations are used to evaluate SM efficiently through implementations. When using the Affine coordinate system with SM, both point doubling and point addition consume 2 multiplications, 1 squaring and 1 division operation in field [48]. Improving on SM leads to reduce computation cost, and running time and thus improve the efficiency of performance in ECC algorithm and its derivatives. The traditional method (double-and-add algorithm) in scalar multiplication uses base 2 in point doubling such as $2(...(2(2P)))$ in addition to point addition [48]. In this method, point doubling is implemented in each bit in k (where k is integer scalar and represented in binary form) while point addition is implemented when bit equal "1" in k [52]. Improving SM makes these algorithms convenient for constrained-source devices such as WSN, RFID, and smart card [14] through reducing of running time and energy consumption. Many researchers have presented methods to represent scalar in order to reduce computation complexity in kP. Many representations have used for SM such as traditional method (double-and-add), double base chain (2,3), multibase representation ((2,3,5) and (2,3,7)) and point halving (1/2P). All these representations lead to shorter representation length of terms and hamming weight. For instance, multibase representation (2,3,5) is shorter terms length and more redundant than a double base chain (2,3). Representing 160-bit in multibase representation (MBNR) needs 15 terms while double base chain (DBNR) needs 23 terms [50]. Sometimes, improving on methods of SM representation may bring storage problems, for example, using point halving instead of point doubling with polynomial base requires greater storage in memory [53]. In the following subsections, we will discuss methods to improve the representation of scalar in SM.

4.1.1.1 Double-Base Chain (2,3)
One of the methods used to develop doubling $(2P)$ is tripling $(3P)$. This method was proposed to use bases 2 and 3 to reduce the execution time of the PM. Ciet et al. [54] proposed a point tripling operation $(3P)$ and mixed it with point doubling $(2P)$ depending on various methods to evaluate $2P + Q$ when 1 inversion is more than 6 multiplications that lead to improving tripling. They presented a comprehensive evaluation through i, s and m for operations types: $P + Q$, $2P$, $2P + Q$, $3P$, $3P + Q$, $4P$, $4P + Q$. The authors noted that $2P+Q$ is faster than $P+(P+Q)$ but $2P+Q$ has slightly more cost. They used the idea of Eisenträger et al. [15] in removing $y3$ from equations when computing $2P + Q$; the authors used this idea with $3P + Q$ and removed $y4$ when 1i is more than 6m to reduce computation cost. Their results proved that this scheme improved SM efficiency in ECC, ECDSA and ECDH. Double-base chain equation is

$$k = \sum_i s_i 2^{b_i} 3^{c_i} \qquad (8)$$

Where s_i is ± 1, and $(2^{b_i} 3^{c_i})$ are integer numbers and b_i and c_i decreased monotonically ($b_1 \geqslant b_2...b_m \geqslant 0$ and $c_1 \geqslant c_2...c_m \geqslant 0$). Furthermore, characteristic 3 was investigated with both Weierstrass and Hessian forms [11]. The authors pointed out that characteristic 3 is efficient in Weierstrass form, where tripling operation performs more efficiently than double and add, while characteristic 3 is not ef-

ficient with Hessian form (Triple-and-add (T&A) used with Weierstrass and double-and-add(D&A) used with Hessian). In addition, the scalar in the double base number system algorithm (DBNS) is analysed on superlinear EC in characteristic 3 [55]. Double-base chain is DBNS but with restriction doubling and tripling and increasing the number of point addition through a Horner-like manner. The authors proposed sublinear SM algorithm for PM (i.e. sublinear in scalar's length) with running time $O(\frac{log\ n}{log\ log\ n})$, and it is faster than D&A and T & A. They pointed out that their algorithm is appropriate to use with large parameters in EC, where selecting large parameters leads to improving performance and security. Also, Dimitrov et al. [56] proposed double-base chains method through prime and binary fields, using point tripling with a Jacobian coordinate in prime fields, and quadrupling combined with quadruple-and-add with an Affine coordinate in binary fields. They used a modified greedy algorithm to convert k to DBNS form. Also, they applied the same idea that was used in Eisenträger et al. [15] to evaluate only x-coordinate in computation $2P+Q$ but with $4P+Q$ to increase SM efficiency. Through results, the authors have become efficient in speeding SM better than classical D&A (21%), NAF (13.5%) and ternary/binary (5.4%) in binary fields. In prime fields, their scheme also gets better results than D&A (25.8%), NAF (15.8%), 4NAF (6%). But results in [54] are better than this scheme in term inversions (with binary fields) in the computation case $4P$ and $4P+Q$, where their scheme obtains for $4P$ ($2[i] + 3[s] + 3[m]$) and for $4P+Q$ ($3[i] + 3[s] + 5[m]$). Results in [54] are $4P(1[i] + 5[s] + 8[m])$ and $4P+Q(2[i] + 6[s] + 10[m])$ where squaring is free and ignored in binary fields, while their scheme is better than [54] when using prime fields.

4.1.1.2 Multibase Representation (2,3,5)
Multibase representation is a method to improve SM through using a point quintupling $(5P)$. It uses 3 bases (2,3,5) called step multibase representation(SMBR). Multibase representation algorithms are shorter terms, more redundant, and have more sparseness than DBNS algorithms. For instance, representation 160-bit costs 23 terms in bases representation (2,3) whereas 15 terms in representation bases (2,3,5). SMBR equation is presented as follows:

$$k = \sum_i s_i 2^{b_i} 3^{c_i} 5^{d_i} \qquad (9)$$

Where s_i is ± 1, and $(2^{b_i} 3^{c_i} 5^{d_i})$ are integer numbers and b_i, c_i and d_i decreased monotonically ($b_1 \geqslant b_2...b_m \geqslant 0, c_1 \geqslant c_2...c_m \geqslant 0$ and $d_1 \geqslant d_2...d_m \geqslant 0$). Mishra and Dimitrov [57] used SMBR with Affine in binary fields and Jacobian in prime fields similar to the scheme in [56]; therefore, their scheme is a generalization for [56] but with 3 bases. They improved on a greedy algorithm (mgreedy) in order to fit SMBR and to gain shorter representation and faster running. Moreover, they recommended using small values for exponents because it does not affect cost. Computation of $5P$ was efficient with the prime field through $2(2P) + P$ and $2P + 3P$. They found that $2(2P) + P$ costs 9s+17m (with Affine) and 14s+20m (with Jacobian) while $2P + 3P$ costs 22m+12s (with Affine) and 26m+16s (with Jacobian). Moreover, computation of $5P$ efficiency with the binary field (Affine) was 1i+5s+13m. Through these results, the authors achieved efficiency in SM with prime and binary fields better than previous algorithms such as D&A, NAF, 3-NAF,4-NAF,

ternary/binary and DB chains whether with precomputation or without precomputation points. Finally, Longa and Miri [58] concluded that the cost of $5P + Q$ is 26m + 13s.

4.1.1.3 Multibase Representation (2,3,7)

Multibase representation is a scalar representation method, which depends on base triple bases to accelerate point multiplication. It uses 3 bases (2,3,7) and is called multibase number representation(MBNR). This representation is the development of previous representation methods (DBNR (2,3) and SMBR (2,3,5)). Purohit and Rawat [50] proposed triple base method (binary, ternary and septenary) with addition and subtraction. The greedy algorithm was used to convert an integer to the triple base (2,3,7), through finding the closest integer to a scalar. Their evaluation referred to this method as having better results than previous methods of scalar representation. They pointed out that septupling ($7P$) costs is less than two formulas ($2(2P) + 3p$ and $2(3P) + P$), where $7P$=3i + 18m, $2(3P) + P$=4i + 18m and $2(2P) + 3P$= 5i + 20m. Also, squaring cost was neglected, regarded as inexpensive in the fields with characteristic 2. MBNR representation was applied through the following equation:

$$k = \sum_i s_i 2^{b_i} 3^{c_i} 7^{d_i} \qquad (10)$$

Where s_i is ± 1, and ($2^{b_i} 3^{c_i} 7^{d_i}$) are integer numbers and b_i, c_i and d_i decreased monotonically ($b_1 \geqslant b_2...b_m \geqslant 0, c_1 \geqslant c_2...c_m \geqslant 0$ and $d_1 \geqslant d_2...d_m \geqslant 0$). MBNR (2,3,7) is better than MBNR (2,3,5) in terms of shorter terms length, more redundant and spare. The next example gives a comparison between using quintaupling and septupling when points number= 895712 as mentioned in [50].

$$(using quintaupling)$$
$$2^4 3^7 5^2 + 2^4 3^5 5^1 + 2^4 3^4 5^0 + 2^1 3^4 5^0 + 2^0 3^2 5^0 + 2^0 3^1 5^0$$
$$(using septupling)$$
$$2^9 3^5 7^1 + 2^7 3^3 7^1 + 2^5 3^1 7^1 + 2^5 3^1 7^0$$

This method presents efficiency in scalar multiplication better than previous representation methods, where we note from the previous example that septupling is more redundant and has fewer terms than quintupling. Also, Chabrier and Tisserand [59] proposed MBNR (2,3,5,7) to represent scalar without precomputation. They investigated the costs in their method when a = -3, the cost is 18m + 11s (prime field). The results of their proposal indicate high performance in FPGA implementations with SM at high speed with storage level and execution time.

4.1.1.4 Point Halving with DBNR and MBNR Representations

This section will discuss point halving (PH) first, then bases (1/2,3), (1/2,3,5), and (1/2,3,7) respectively. Point halving method is one of the methods used to reduce the cost of point doubling. It means extract P from $2P$ [48]. Knudsen [53] and Schroeppel [60] separately proposed point halving in order to accelerate point multiplication. They wanted to reduce the cost of computation complexity in point doubling. Point halving was suggested to use instead of all point doubling, where point halving is faster than point doubling in the case of using Affine coordinate on the curve (minimal two-torsion) [53]. This method was implemented on the binary field (polynomial and normal). Both polynomial and normal basis perform fast

computations, but polynomial basis suffers from storage problem. This scheme neglected squaring operation and evaluation as it depends on inversion and multiplication. To implement point halving on binary field in curve, the following equations were used to get $Q = 2P$, where $P = (x, y)$ and $Q = (u, v)$:

$$\lambda = x + \frac{y}{\lambda} \qquad (11)$$

$$u = (\lambda)^2 + \lambda + a \qquad (12)$$

$$v = x^2 + u(\lambda + 1) \qquad (13)$$

Through previous equations, we get point halving; the second equation gives us λ value, the third equation gives us x and subsequently uses values of x, λ to get y value from the first equation. Fong et al. [61] presented analysis and comparison between SM methods (point doubling and point halving) and used binary fields ($F_{2^{163}}$ and $F_{2^{233}}$) through reduction polynomials of trinomial and pentanomial (using polynomial basis instead of normal) on standards of NIST's FIPS 186-2. Through analysis, they found that point halving is faster (29%) than point doubling when P is known in advance. Also, they noted τ-adic (Frobenius Endomorphism) used in [62] faster than PH. They presented a comparison between double-and-add and halve-and-add (H&A) over $F_{2^{163}}$ through Affine and Projective coordinate systems. Also, they pointed out that point doubling can use with mixed coordinate while point halving must be used in the Affine coordinate. They explained that signature verification in ECDSA has 2 SM and this operation is expensive when P is known. But performing ECDSA verification with halving is more efficient than doubling in addition to halving being better for constrained-resources in the case assumption that storage is available. Ismail et al. [48] mentioned that point halving is faster than point doubling by 5-24%.

Point halving ($1/2P$) is combined with DBNR (2,3) in [52] to reduce computation complexity and to increase operations efficiency of scalar multiplication in ECC/ECDSA. The authors implemented their scheme on Pentium D 3.00 GHz using C++ with MIRACL library V 5.0 that deals with a large integer. Equations of scalar representation in this scheme are:

$$k' = 2^q mod p \qquad (14)$$

Where 2^q a large integer and value it close to field size.

$$k = k'/2^q = \frac{\sum_{i=1}^m s_i 2^{b_i} 3^{c_i}}{2^q} = \sum_{i=1}^m s_i (1/2)^{q-b_i} 3^{c_i} \qquad (15)$$

Where s_i is ± 1, and ($2^{b_i} 3^{c_i}$) are integer numbers and b_i and c_i decreased monotonically ($b_1 \geqslant b_2...b_m \geqslant 0$ and $c_1 \geqslant c_2...c_m \geqslant 0$). The authors showed a comparison between their scheme results and DBNR with using the binary field (163-bit, 233-bit, and 283-bit) and prime field (192-bit) according to NIST's standards. Their scheme achieved better results than an original double-base chain, their results reduced 1/2 inversion, 1/3 squaring and few number of multiplication and that led to improving DBNR in scalar multiplication.

In addition, Ismail et al. [48] presented improving on SMBR method through using point halving with SMBR. The original algorithm used bases (2,3,5) while the modified algorithm used (1/2,3,5). The modified scheme adopted point halving and halve-and-add instead of point doubling and double-and-

add. It applied the following equation:

$$k = k'/2^q = \frac{\sum_{i=1}^{m} s_i 2^{b_i} 3^{c_i} 5^{d_i}}{2^q} = \sum_{i=1}^{m} s_i (1/2)^{q-b_i} 3^{c_i} 5^{d_i}$$

(16)

Where s_i is ± 1, and $(2^{b_i} 3^{c_i} 5^{d_i})$ are integer numbers and b_i, c_i and d_i decreased monotonically ($b_1 \geqslant b_2 ... b_m \geqslant 0, c_1 \geqslant c_2 ... c_m \geqslant 0$ and $d_1 \geqslant d_2 ... d_m \geqslant 0$). The greedy algorithm is used to convert an integer to modified MBNR. The authors referred to MBNR (2,3,5) and modified MBNR(1/2,3,5) as having the same terms but costs in modified MBNR (i=7,s=17,m=77) were less than original MBNR (i=15,s=36,m=80) with k = 314159. Sometimes, the original algorithm has fewer terms than the modified MBNR, but modified algorithm remains less costly in operations(i,s,m). The modified and original algorithm are applied on different curves sizes (163, 233, 283), where each curve is tested with 100 random numbers. Results refer to modified scheme of less computation cost (30%) rather than the original scheme (2,3,5).

The authors in [51] proposed combining point halving with MBNR (2,3,7). They used bases (1/2,3,7) instead of (2,3,7). This scheme is less costly than schemes that have bases (2,3,5) and (2,3,7). Also, this scheme is used with the binary field. The authors referred to MBNR as being convenient for ECC because of its shorter representation length and less hamming weight. MBNR (1/2,3,7) representation uses the following equation:

$$k = k'/2^q = \frac{\sum_{i=1}^{m} s_i 2^{b_i} 3^{c_i} 7^{d_i}}{2^q} = \sum_{i=1}^{m} s_i (1/2)^{q-b_i} 3^{c_i} 7^{d_i}$$

(17)

Where s_i is ± 1, and $(2^{b_i} 3^{c_i} 7^{d_i})$ are integer numbers and b_i, c_i and d_i decreased monotonically ($b_1 \geqslant b_2 ... b_m \geqslant 0, c_1 \geqslant c_2 ... c_m \geqslant 0$ and $d_1 \geqslant d_2 ... d_m \geqslant 0$). This scheme presents improvement in the performance of scalar multiplication through reducing computation complexity. Through these results, the authors pointed out that their scheme reduced inversion to 1/2, and squaring to 1/3 and there were fewer numbers of multiplication when compared with previous schemes. Table 3 shows the costs of arithmetic operations to represent scalar.

4.1.2 Methods of Improving Curve Operations

In this section, we describe the methods which are used to improve the PM by reducing the addition formula operations (point addition and point doubling). In the second part, we explain that some research efforts have used the different methods for first part methods.

4.1.2.1 Improvement Via Various PM Methods

Many approaches can be used to improve PM methods because the PM operation consumes a large amount of time from ECC and ECDSA algorithms. Many researchers have presented different methods to improve the performance of basic PM algorithm (D&A) such as NAF, Window, Comb, Montgomery, Frobenius. In this section, these algorithms will be elaborated. Table 4 shows improvements on SM methods.

- **Double-and-Add (D&A) Method**
  D&A is a basic algorithm in PM. It has been repeatedly used for two operations: point addition($P + Q$) and point doubling ($P + P$). This algorithm is similar to the

---

Algorithm 2 The D&A algorithm

Input: point P on $E(F_p)$; l-bits scalar $k = (k_{l-1}...k_0)_2$
Output: $Q = k.P$
$Q = P$
for i = l - 2 downto 0 **do**
    $Q = 2.Q$
    if $k(i)$= 1 **then** $Q = Q + P$
return Q

Algorithm 3 NAF algorithm:

Input: A positive integer k
Output: NAF(k)
i=0
while $k \leq 1$ **do**
    if k is odd **then**
        k_i=2$-$(k mod 4)
        $k = k - k_i$
    else
        $k_i = 0$
        $k = k/2$, $i = i$ +1.
return ($k_i - 1, k - i - 2, ..., k_1, k_0$).

---

square-and-multiply algorithm [13], and depends on bits in scalar k after it was represent it in binary form, where if bit in k has zero value, the D&A algorithm will execute point doubling, while if bit in k has one value, this algorithm will execute the point doubling and point addition in each loop. In this method, the number of point addition is a half of the point doubling. D&A method is explained in algorithm 2.

- **Non Adjacent Form (NAF) Method**
  NAF is the representation of signed-digit and was introduced to reduce execution time in PM; it outperforms D&A algorithm. It does not allow for any two nonzero bits in scalar k to be adjacent [13, 52] and this leads to reducing hamming weight as PM computation depends on the number of zeros and bits length in a scalar. As a result, this algorithm reduces the number of point addition. The NAF method is faster than the square-and-multiply method [19]. This method can reduce the number of point addition to one-third. It is represented using the following equation:

$$NAF(k) = \sum_{i=0}^{l-1} k_i 2^i$$

(18)

Where $k \in \{0, \pm 1\}$, NAF method is explained in algorithm 3 [63]. When NAF is combined with a window through a random point in PM with the Jacobian coordinate system, NAF can achieve an improvement of 33.7 addition and 157.9 doubling (with k size is 160-bit) [64]. Also, Gallant et al. [65] combined simultaneous multiple point multiplication with left-to-right window-NAF, where they convert form (kP) for PM to form $k_1 p + k_2 \phi(p)$ (ϕ which is Endomorphism operation). This method improved point doubling (79 points) and point addition (38 points) that lead to improving PM to 50% better than traditional methods with F_{p160}. Similarly, Mishra and Dimitrov [57] made a comparison between their scheme (Quintupling) and NAF algorithm (in addi-

Table 3. Costs of inversion (i), squaring (s), and multiplication (m) for scalar representation

Scalar representation	Binary field			Prime Field		
	i	s	m	i	s	m
1/2P	-	-	1	-	-	-
1/2 P + Q	1	-	5	-	-	-
P +Q	1	1	2	1	1	2
2P	1	1	2	1	1	2
2P + Q	1	2	9	1	2	9
3P	1	4	7	1	4	7
3P + Q	2	3	9	2	3	9
4P	1	5	8	1	9	9
4P + Q	2	6	10	2	4	11
5P	1	5	13	-	10	15
5P + Q	-	-	-	1	13	26
7P	3	7	18	-	11	18
7P + Q	-	-	-	1	22	28
1/2 & DBNR	DBNR/2	DBNR/3	Slightly lower than DBNR	DBNR/2	DBNR/3	Slightly lower than DBNR
1/2 & SMBR	SMBR/2	SMBR/3	Slightly lower than SMBR	SMBR/2	SMBR/3	Slightly lower than SMBR
1/2 & MBNR	MBNR/2	MBNR/3	Slightly lower than MBNR	MBNR/2	MBNR/3	Slightly lower than MBNR

Algorithm 4 Frobenius with D&A algorithm:

Input: Point P, τ-adic expansion of k $(k_{l-1}, ... k_0)$.
Output: $k.P$
$P_1 = P$ and $P_2 = 2P$
for i in 0 to (l-1) loop **do**
    if $k(i) = 1$ **then**
        $Q = Q + P$
    else if $k(i) = -1$ **then**
        $Q = Q - P$
    $P = \text{Frobenius}(P)$
End loop

---

tion to many NAF versions such as NAF-3 and NAF-4). Furthermore, NAF algorithm (right-to-left binary method) without precomputations (to use in constrained devices) was used with the mixed coordinate system to improve PM [9]. The author compared his algorithm with NAF (left-to-right binary method). Results showed that this scheme presented better performance than NAF (left-to-right binary method). Also, Purohit et al. [51] pointed out that multibase non-adjacent form (mbNAF) provided a speeding up of the execution time in PM with improving scalar performance through multibase.

- **Frobenius Map Method**
  Some researchers have used Frobenius map algorithm instead of point doubling ($2P$) because Frobenius map performs squaring operations $\tau(x, y) = \tau(x^2, y^2)$ where τ is Frobenius [62] (that is replacing point doubling with 2 squaring), and is represented by the following equations:

$$2.P = -\tau^2 P + \mu\tau P \qquad (19)$$

Where $\mu = \pm 1$, and

$$k = \sum_{i=0}^{t-1\tau} s_i \tau^i \qquad (20)$$

Where s_i is 0 or ± 1 and τ is Frobenius endomorphism. Scalar k value is obtained through the division repeated for s by τ [14] as in the foregoing equation. Algorithm 4 shows combining Frobenius map instead of point doubling with D&A; this algorithm uses addition, subtraction, and Frobenius. Frobenius endomorphism (τ-adic) was combined with point halving [62]. The authors used point halving because it is three times faster than point doubling; therefore, they used τ-and-add instead of

double-and-add with Koblitz curves; these curves have useful features in acceleration kP. They used these curves with fields sizes (k-163 bit and k-233 bit) to standards of NIST's FIPS in binary fields. τ-NAF was used to reduce point addition from $n/3$ to $2n/7$. Results have proved that τ-and-add based τ-NAF provides speeding 14.28% better than Frobenius method, and in addition, it does not require of additional memory in the case using of normal basis. Frobenius expansion method for the special hyperelliptic curve (introduced by Koblitz) with GLV(Gallant-Lambert-Vanstone) endomorphism (using fields of large characteristic) is presented to improve the efficiency of scalar multiplication [66]. Through the results, the authors largely improved point doubling while point addition did not improve. This scheme improved scalar point multiplication from 15.6 to 28.3% when it was implemented on F_{p^n}. Reducing the time and increasing performance in ECDSA algorithm is performed through exchange doubling point with Frobenius scheme in the PM [14]. The authors presented ECDSA algorithm with curves of subfields Koblitz for binary fields (key length a 163-bit). They explained that binary fields are convenient for hardware implementation. This scheme achieved good results in performance over time where key generation took time 0.2 ms, signature generation took 0.8 ms and signature verification took 0.4 ms. In addition, this scheme is suited for constrained-source devices such as smart cards, WSN and RFID. In 2017, Liu et al. [67] submitted a study of the application of twisted Edwards curve (considered efficient models of ECC/ECDSA) with Frobenius endomorphism to reduce point doubling to 50% for traditional algorithms. They used the 207-bit key (prime field) with two hardware architectures (ASIC and FPGA). They pointed out that their method offers high performance, less memory, and time requirements compared to traditional Frobenius algorithms. Also, they mentioned that integrating their method with a window of 2 size would save 1/16 of the point addition.

- **Window Method (WM)**
  Window algorithm is intended to improve execution time in D&A method through specific window size (terms computation in a scalar); if window size equals one, the window algorithm is the same as D&A algorithm. This

Algorithm 5 Window algorithm:

Input: point P on $E(F_p)$, Window width w, $d = [l/w]$, $k = (k_{d-1}, ..., k_0)_2 w$.
Output: $Q = kP$.
$P_0 = P$
Precompute: for i = 1 to 2^{w-1} do: $P_i = P_{i-1} + P$
$Q = 0$
for i = d-1 downto 0 **do**
    $Q = 2^w Q$
    $Q = Q + P_{kd}$
return Q

algorithm uses precomputation for points in the case of the fixed point multiplication (FPM) and also reduced point additions better than D&A [13, 46], as explained in algorithm 5. Wang and Li [24] used NAF and window methods to improve performance for PM algorithm. They found that window method is more efficient than NAF. They used hybrid multiplication to reduce access memory instead of multi-precision multiplication. For a practical example, ECC results on MICAz is signature generation=1.3s and signature verification = 2.8s while on TelosB is signature generation=1.6s and signature verification = 3.3s. The authors showed the possibility of using ECC on WSN. Also, the variable-length sliding window was combined with NAF method to reduce points addition (PM) in ECC and ECDSA algorithms [19]. Computation complexity in PM depends on bits' length and zeros number in an integer. The authors divided elements in NAF(k) to two windows (non-zero and zero windows) and also divided non-zero window to six types in sliding window. In addition, they used coordinate system (Jacobian) with point doubling and coordinate system (Jacobian Affine) with point addition. Their results demonstrate that their scheme is better and more efficient than NAF, wNAF and square-multiply schemes in terms of efficient point multiplication and time of public key generation.

- **Comb Method (CM)**
  This method uses binary matrix (w, d) to compute FPM efficiently, where w is row and d is column. It was introduced by Lim and Lee and this algorithm is case special from multi-exponentiation using Straus' trick [68]. Comb method uses precomputation to improve PM performance [13, 63]. as in algorithm 6. Comb method was modified in [69] to improve PM through combining comb method with width-w NAF. This method is essentially designed to reduce the number of point addition. It has presented better results than original comb and NAF with comb. Also, it is used to accelerate multiple PM $(kP + rQ)$ in ECDSA algorithm. The authors explained that their algorithm is convenient for devices constrained-source (memory) when choice suited parameters for an algorithm such as window size. It improved computation complexity from 33% to 38% compared to NAF with CM in devices constrained-sources.
- **Montgomery Method**
  Montgomery algorithm eliminates division operation and uses reduction operation efficiently [24]. This algorithm was introduced by Montgomery and uses only x-

Algorithm 6 Comb algorithm:

Input: A point P, an integer k, and a window width $w > 2$
Output: $Q = kP$.

Precomputation Stage:
Compute $[b(w - 1), ..., b1, b0] P$ for all $(b(w - 1), ..., b1, b0) \in Z_2^w$
Write $K = (K(w-1)||...||K^1||K^0)$ padding with 0 on the left if necessary, where each K^j is a bit-string of length d.
Let K_i^j denote the i-th bit of K^i
Define $K_i = [K_i^{w-1}, ..., K_i^1, K_i^0]$
$Q = O$

Evaluation Stage:
for $i = d$-1 downto 0 **do**
    $Q = 2Q$,
    $Q = Q + k_i P$
return Q

Algorithm 7 Montgomery ladder algorithm:

Input: A point P on E and a positive integer $k = (k_{l-1}...k_0)_2$.
Output: The point kP.

$P_1 = P$ and $P_2 = 2P$
for $i = l - 2$ down to 0 **do**
    if $k_i = 0$ **then**
        $P_1 = 2P_1$ and $P_2 = P_1 \oplus P_2$
    else
        $P_1 = P_1 \oplus P_2$ and $P_2 = 2P_2$
return P_1

coordinate and removes y-coordinate, and this leads to increasing PM performance.

A method introduced by Eisenträger el al. has improved PM's performance with an Affine coordinate in ECC [15]. This method eliminates field multiplication in the case of using left-to-right of binary SM. It eliminates y-coordinate in addition, doubling and tripling operations. This scheme led to saving field multiplication and thus achieved an improvement of PM's cost from 3.8% to 8.5%. For example, in order to perform the operation of $2P + Q$, authors used form $((P + Q) + P)$; also in the operation of $3P + Q$, they used $(((P + Q) + P) + P)$. This trick led to the improvement in PM operation. For instance, with a number of points (1133044P), the cost of original PM is 23i+41s+23m, while the cost of improved PM is 23i+37s+19m, namely, an improvement in m and s. Ciet and Joye [54] used Eisenträger's idea but with $3P + Q$ when 1i equal more than 6m to reduce the computation cost of PM. Also, Dimitrov et al.[56] used Eisenträger's idea but with $4P + Q$ to improve the PM efficiency. Montgomery method is represented in algorithm 7. Saqib et al. [70] suggested that using Montgomery algorithm depends on the parallel-sequential manner to accelerate PM in ECC with $GF(2^{191})$ on Xilinx VirtexE XCV3200 FPG device. Their scheme improved PM's performance to $56.44\mu s$. They compared their scheme with previous schemes (different H/W de-

vices). Results showed that this scheme presents better performance. Also, they pointed out the balance of their scheme between memory size and time. It was suggested to use the Montgomery algorithm with ECDSA (with keys length 224-bit and 256-bit) that has high costs in both computation and communication [71]. The authors analysed time complexity, where PM consumes a large amount of time from ECDSA's time. They pointed out that Montgomery offers better performance than other algorithms in constrained environments and mobility. Hutter et al. [47] used Montgomery ladder algorithm to point multiplication in ECC (with randomized Projective coordinate) because it provides security against several attacks and performs all operations with x-coordinate, thereby increasing PM performance through implementation of ASIC processor with asymmetric cryptography ECDSA (F_{p192}). In 2017, Liu et al. [72] adopted the Montgomery method with the lightweight elliptic curve (twisted Edwards curve (p159, p191, p223, and p255)) to improve speed and balance between memory and performance (cost of communication, execution time, memory) as well as security requirement. They implemented the ECDSA algorithm on Tmote Sky and MICAz nodes. During the implementation, they noted that their scheme offers memory efficiency which requires 6.7k bytes for a SM process instead of 13.1 kbytes for the traditional Montgomery scheme. They recommended the exact selection of ECDSA's parameters curves and the balance between security and efficiency requirements.

In summary, the best algorithm for PM (ECC and ECDSA) it is fixed-base comb with w=4 in random binary curves, while in Koblitz curves it is fixed-base window TNAF (τ-adic NAF) with w=6 in case memory is not constrained [73]. When memory is constrained, Montgomery is the best with random binary and TNAF is the best with Koblitz. The authors pointed out that Koblitz curves are faster than random binary curves ($F_{2^{163}}$, $F_{2^{233}}$ and $F_{2^{283}}$) for NIST's standards [73]. Moreover, Rafik and Mohammed [13] analysed in detail the types of SPM (scalar point multiplication) algorithms (that are D&A, window, and comb method). Examining the results, they concluded that the CM method is faster than D&A and WM because the CM uses less doubling and adding points, but this method needs more memory. As for security, the D&A method is the best with 27%. D&A is good at memory requirements, needs more time. Therefore, they concluded that several concepts of SPM in ECC algorithm (or ECDSA or ECDH) are based on the user's application and constrained sources. If execution time is a large, the energy consumption increases. The authors carried out the Secure-CM and earned good time to implement (1.57 s); as demonstrated ECC is applicable in WSN.

4.1.2.2 *Improvement of Efficiency Via Other Methods*

In this subsection, we will explain a set of different ideas from previous methods to improve ECDSA performance through reducing consumption time in signature generation and signature verification.

A method was suggested to accelerate signature verification in ECDSA algorithm [74]. The authors used a number of small bits (side information (1 or 2 bits)) to specify the number of allowed points in point multiplication; through the modified

algorithm, points double was largely reduced. They implemented traditional ECDSA algorithm and modified ECDSA algorithm on ARM7TDMI platform processor with a finite field F_{p384} by NIST, and achieved the result that modified ECDSA verification is 40% better than traditional ECDSA. They discovered that this changing does not affect ECDSA standards and proved that their scheme has the same security as traditional ECDSA. Also, an addition formula was proposed through using Euclidean Addition Chains (EAC) with Fibonacci to avoid difficult to find small chain [75]. The author has used a Fibonacci-and-add algorithm instead of double-and-add (using Jacobian coordinate and characteristic greater than 3). Some improvements were added to this scheme through the window and signed representation to improve algorithm performance. The author has compared his scheme with many schemes (D&A, NAF, 4-NAF and Montgomery ladder). Results indicated that his scheme outperforms D&A only in the case of improvement addition (window or signed representation). In addition, the improvement of signature verification in ECDSA is achieved through cooperating of adjacent nodes in the computation of intermediate results [27]. The proposed scheme used 1PM+1Add in nodes that use intermediate results instead of 2PM+1Add as in the original scheme. The authors analysed performance and security in the scheme of signature verification with many attacks (independent and collusive). They noted that these attacks do not have a large effect on their scheme. The modified scheme has saved energy consumption 17.7-34.5% better than the traditional scheme, as signature verification in the modified scheme is 50% faster than the original scheme. They implemented their scheme on Micaz motes with the finite field ($F_{2^{163}}$).

Furthermore, Li et al. [46] proposed a scheme to improve scalar multiplication by reducing point additions. They presented a method to generate k depend on the generation of an integer S periodically. They achieved good results in improving point addition $\sqrt{3/4 \times L}$ whereas binary scheme ($L/2$). Point doubling in their scheme is similar to previous schemes (D&A, NAF, WM). Their scheme does not require additional memory and depends on the growth rate of a small with bit length growth in scalar(k). Also, it is appropriate to implement in H/W because it needs simple operands. Because scalar multiplication (ECDSA) consumes a large time through processing, that leads to power consumption [76]. The authors exchanged linear point multiplication method with their method using divide and conquer algorithm. This algorithm uses a binary tree to quotient values and a skew tree to reminder values, where it processes points through parallel computation method. Results showed that they achieved better efficiency than linear scalar multiplication in terms of a number of clock pulses and power consumption. In 2017, Guerrini et al. [77] have proposed a method of random scalar that relies on the covering systems of congruence relationships. A random was applied in a scalar representation using mixed-radix SM algorithm. They generated n-covers randomly with a greedy method, depending on available memory as well as recommended selection of a large n. Also, they pointed out that their method is more efficient, less expensive and incurs fewer additional expenses for arithmetic operations than the Coron's randomization, D&A, NAF, and wNAF methods. Ta-

ble 4 shows improvements on other SM methods.

4.2 Efficiency Improvement of Coordinate Systems

ECDSA has to perform complex operations; these operations consume resources in constrained devices. Using the appropriate coordinate system, could lead to reducing costs of high computation in doubling and addition operations. The improvement of computation complexity in PM depends on point representation that is considered important in curve operations [78]. Many different coordinate systems used with these algorithms such as Affine, Projective, Jacobian, Chudnovsky Jacobian, modified Jacobian, and mixed. Coordinate systems are used to represent and contribute to speeding computation in ECC/ECDSA algorithm. There is no coordinate system to accelerate both point addition and doubling [3]. Some coordinate systems require faster or slower computation time in the point doubling and point addition operations than other coordinate systems; for instance, point doubling uses less computation in Jacobian while the point addition uses more computation in the Jacobian than Projective and Chudnovsky Jacobian coordinates.

- **Using Affine Coordinate**

  Affine is a basic coordinate system that is used with doubling and addition operations. This system uses two coordinates (x, y). When point addition $(P + Q)$ uses the Affine system, then the cost of the resulting point (R) is 1 inversion, 2 multiplication and 1 squaring, while cost(R) in the case point doubling is 1 inversion, 2 multiplication, and 2 squaring [79]. Costly arithmetical operations in point addition and point doubling are multiplication, squaring and inversion. Inversion operation is much lower than multiplication operation. The Affine coordinate system uses inversion operations [24]. Inversion operation is required in point addition and point doubling when using Affine coordinate. But Affine coordinate needs fewer multiplication operations than other coordinate systems such as Projective coordinate [64].

- **Using Projective Coordinate**

  Inversion operation is dramatically expensive in point addition and point doubling. The Projective coordinate system removes this operation and that leads to improving PM performance in ECC/ECDSA. This system uses three coordinates (X, Y, Z) and $Z \neq 0$ [79], which take formula $(X/Z, Y/Z)$ corresponding to Affine coordinate. The Projective coordinate is represented as the following equation (with prime fields) [13]:

  $$Y^2 Z = X^3 + aXZ^2 + bZ^3 \qquad (21)$$

  López and Dahab also used three coordinates but using formula $(X/Z, Y/Z^2)$ and $Z \neq 0$ corresponding to Affine coordinate [80]. Their scheme improved original Projective on a curve in the binary field. Furthermore, Projective coordinate is used with random binary and koblitz curves ($F_{2^{163}}$, $F_{2^{233}}$ and $F_{2^{283}}$) for NIST's standards instead of Affine [73]. This coordinate leads to significant improvement in PM ECC and ECDSA. Then, Hutter et al. [47] used randomized Projective coordinate through generating a randomized number on x-coordinate in base point. Also, different coordinate systems have been discussed, such as Projective and Affine coordinates [13]. Through experiments, the authors found that Projective coordinates are faster by 91%

than Affine coordinates but Projective coordinate avoids inversion operation, and vice versa for the memory, while the Projective coordinates require more memory because they use three coordinates (X, Y, Z). Oliveira et al. [78] proposed λ-Projective coordinate through using three coordinates (X, Y, Z) with binary field ($F_{2^{254}}$) to improve computation efficiency in PM. The formula used for coordinates is $(X/Z, L/Z)$ corresponding with λ-Affine formula, where the λ-Affine formula is $(X, X + Y/X)$. λ-Projective improved López and Dahab Projective(LD-Projective), where the $P + Q$ cost in λ-Projective is 11m+2s while in LD-Projective it is 13m+4s. Also, $2P + Q$ cost in λ-Projective is 10m+6s while in LD-Projective is 11m+10s. In addition, λ-Projective improved H&A to 60% in squaring and 6% in multiplication better than LD-Projective. Their scheme was implemented on Sandy Bridge platform. The authors achieved computation 69500 clock cycles through the single core with H&A and 47900 clock cycles through the multi core combining Gallant, Lambert, and Vanstone, (GLV) technique, H&A and D&A in SM unprotected. Also, they achieved computation 114800 clock cycles through single core protected. Their scheme achieves performance better by 2% than Ivy Bridge and 46% than Haswell platforms. In 2017, Al-Somani [81] proposed a parallel SM method based on López-Dahab Projective that required 4 multiplications for DBL and 16 multiplications for ADD. His scheme applied on FPGA and AT^2 processors and obtained high speed for SM's performance when this scheme was implemented on eight processors with López-Dahab Projective.

- **Using Jacobian Coordinate**

  This coordinate system does not require inversion in addition formula (addition and doubling), but uses inversion only in the final computation stage. It is similar to Pojective coordinate (also it is an improvement of Projective coordinate). Jacobian provides less running time in point doubling and more running time in point addition than Projective coordinate [64]. This system uses three coordinates (X, Y, Z) and $Z \neq 0$, which take formula $(X/Z^2, Y/Z^3)$ corresponding for Affine coordinate [79].

  $$Y^2 Z = X^3 + aXZ^4 + bZ^6 \qquad (22)$$

  Due to the point addition, Jacobian consumes more computation time than other coordinate systems. Chudnovsky Jacobian coordinate is proposed to accelerate point addition in the Jacobian coordinate. This system uses coordinates (X, Y, Z, Z^2, Z^3) and saves 1m+1s in point addition [82]. This coordinate has a disadvantage that point doubling is slower than Jacobian coordinate [9]. The author pointed out that Chudnovsky is the fastest in point addition and modified Jacobian is the fastest in point doubling. A Jacobian coordinate is used to improve running time in EC exponentiation with fixed point and random point. This coordinate is better than Affine and Projective coordinates [64]. Also, modified Jacobian coordinate system was able to achieve better performance than Affine, Projective, Jacobian and Chudnovsky Jacobian coordinate systems, where Affine has addition cost (i+2m+s) and doubling cost(i+2m+2s), Projective has addition cost (12m+2s) and doubling cost

Table 4. Improvements on different SM methods

Paper	SM methods	Subsequent improvements of SM (PA and PD)	Field
[19]	Traditional NAF	1/3 PA	F_p
[64]	wNAF	Random point: PA (33.7) and PD (157.9) Fixed point: PA (30) and PD (15)	F_{p160}
[65]	wNAF with Endomorphism	Roughly 50% SM	F_{p160}
[51]	mbNAF	50% SM	F_p and F_{2^m}
[13]	Traditional D&A	PA is 1/2 PD	F_p
[62]	τ-NAF τ-NAF with PH	1/3 PA and reduce PD 2n/7 PA and reduce PD	F_{2^m} (163,233)
[66]	Frobenius with GLV	28.3% SM	F_p
[67]	Frobenius with twisted Edwards curve	3n/4 PA and 1/2 PD	F_{p207}
[46]	Traditional window	Reduce PA and PD compared with NAF	F_p (192,256,512)
[24]	Sliding window	10% better than NAF	F_{p512}
[19]	Variable-length sliding window	27.4% better than wNAF	F_p
[13]	Traditional cm	cm is less PA and PD than wm and D&A	F_p
[69]	cm with NAF cm with wNAF	33% SM 38% SM	F_{p160}
[15]	Traditional Montgomery	8.5% SM memory size (SM)=13.1k	F_p (160,256)
[56]	Improved Montgomery	Prime field: 21% SM is better than Q & A 13.5% SM is better than NAF Binary field: 25.8% SM is better than Q & A 15.8% SM is better than NAF	F_p and F_{2^m}
[54]	Improved Montgomery	Reduce PA and PD in traditional Montgomery	F_p
[70]	Montgomery with parallel sequential	roughly 51% SM with time=0.056ms	$F_{2^{191}}$
[72]	Montgomery with twisted Edwards curve	Faster SM than traditional Montgomery memory size (SM)=6.7k	F_p (159,191,223,255)
[74]	SM with side information	40% is better than traditional verification	F_{p384}
[75]	SM with Fibonacci	Improve PA in D&A and reduce PA cost (10%)	F_{2^m}
[27]	SM with intermediate results	Improve signature verification (50%)	$F_{2^{163}}$
[46]	SM with integer S periodically	Improve PA ($\sqrt{3/4} \times L$ is better than D&A, NAF, and WM	F_p (192,256)
[76]	SM with binary tree	Reduce SM time and complex computations	F_{2^m}
[77]	SM with random n-cover	Reduce SM cost is better than D&A, NAF, and wNAF	F_p

(7m+5s), Jacobian has addition cost (12m+4s) and doubling cost (4m+6s), Chudnovsky Jacobian has addition cost(11m+3s) and doubling cost (5m+6s) and, modified Jacobian has addition cost (13m+6s) and doubling cost (4m+4s) [3]. This coordinate system is applied on EC exponentiation through prime field (F_{p160}, F_{p192} and F_{p224}). Modified Jacobian is faster point doubling than all previous coordinate systems. In addition, the modified Jacobian coordinate is used to represent addition formula when ECDSA algorithm is implemented as authentication protocol in wireless on processor ARM7TDMI [2]. The authors implemented ECDSA with F_p with different field sizes (F_{p160}, F_{p176}, F_{p192}, F_{p208} and F_{p256}) depending on standards of ANSI X9F1 and IEEE P1363. Through results, they get running time 46.4ms for signature generation and 92.4ms for signature verification when using F_{p160}. They pointed out that their scheme improved bandwidth and storage compared with previous schemes. Brown et al. [79] pointed out that Jacobian is faster than Affine, Projective, and Chudnovsky in point doubling. Through results, these coordinate systems improved speeding of computation in ECC/ECDSA through reducing execution cycle.

- **Using Mixed Coordinate Systems**

The mixed coordinate system uses more one coordinate system to represent addition formula (which each point

uses a different system) [3, 9] to get the best performance and least computation time in PM. Many different mixed coordinate systems such as Jacobian-Affine, Affine-Projective, and Chudonvsky-Affine [79].

Cohen et al. [3] pointed out that modified Jacobian mixed with other coordinate systems (Jacobian and Affine or Jacobian and Chudnovsky Jacobian) presents a significant improvement on computation time with the prime field (F_{p160}, F_{p192} and F_{p224}). Execution time is reduced using Jacobian-Chudnovsky coordinate with prime field (when using w=5 for F_{p192}, F_{p224} and F_{p256}, and w=6 for F_{p384} and F_{p521}). Also, Jacobian and Chudnovsky coordinates improve PM performance and Chudnosky is preferred although that requires some extra storage (precomputation for points) [79]. Different coordinate systems were investigated (Affine, Projective, Jacobian and LD Projective) on point addition and doubling with characteristic 3 in both Weierstrass and Hessian forms [11]. The authors noted that Jacobian is the most efficient with Weierstrass while Projective is the most efficient with Hessian. Also, they used mixed coordinate (Affine-Projective) on both Weierstrass and Hessian. Through using mixed coordinate, results indicate improved timing in addition formula. Mixed coordinate system (Affine-Jacobian) is used to reduce m and s operations or to avoid i operations [24]. Authors pointed out that mixed co-

ordinate is better performance (6%) than Jacobian co-ordinate. In the mixed coordinate, point addition consumes 8m and 3s (12 modular reductions) while point doubling consumes 4m and 4s (11 modular reductions). Also, mixed coordinate is used with right-to-left NAF algorithm [9], where applied Jacobian is used with point addition and modified Jacobian with point doubling. The author pointed out that modified Jacobian is fast in point doubling but is slow in point addition; therefore, the author used Jacobian in point addition. Through results, this scheme presented efficiency of a scalar by $13.33\ell m$ (where ℓ is bits' length in a scalar) compared with Jacobian ($15.33\ell m$) and modified Jacobian ($14.33\ell m$). But when three fields (temporary) variables additional are used, then the left-to-right binary method described in [3] was a better performance. Jacobian-Affine is faster than other coordinate systems in point addition. Using Jacobian coordinate in point doubling and Jacobian-Affine in point addition [19]. Authors pointed out that Jacobian is the fastest in point doubling and mixed coordinate is the fastest in point addition. Oliveira et al. [78] used mixed coordinate to compute $2P + Q$ where λ-Affine was used to represent P point and λ-Projective to represent Q point. They achieved computation cost less than LD-Projective. In 2017, Pan et al. [83] indicated that mixed Jacobian-Affine coordinate is more efficient than Jacobian coordinate. They applied a mixed coordinate with ECDSA to improve the efficiency of computation processes to authenticate users during online registration. Their results indicated that mixed coordinate reduces calculations in both point addition and point doubling. Table 5 shows the successive improvements of different coordinate systems in SM such as Affine, Projective, Jacobian, and mixed coordinates.

4.3 Efficiency Improvement Via Algorithm Simplification

This section will show some different approaches to improving ECDSA performance. A set of ideas which researchers used to simplify the algorithm to improve the performance ECDSA includes removing the inversion in the generation of the signature and the signature verification, a collection of signatures, using a few keys and using two points instead of shared three curved points.

ECDSA algorithm is mainly used to provide integration, authorisation, and non-repudiation depending on servers' CA (certification authority) [84]. The authors added an ID, ECDSA threshold and trust value with the public key to authorise node to gain access to the information. Also, it was stated that data transmission of ECDSA in the wireless sensor network (WSN) consumes more energy than computation operations [85]. To reduce energy consumption, in data transmission, all encrypted data aggregated from all sensor nodes are directly sent to the Base station without decryption. This operation leads to reducing energy consumption and increasing network lifetime. Encryption and signing algorithms are used for the collected data [86, 85]. Encryption algorithm EC-OU (Elliptic Curve Okamoto-Uchiyama) is used to maintain confidentiality and ECDSA algorithm to maintain integration, both algorithms are used during homomorphic. The idea behind this scheme is that encryptions, signatures and public

keys gathered together from all nodes are sent to aggregator (CH), then these aggregators are forwarded to the top level (parent aggregators) and so on up BS, that is, it is not the decryption or signature verification in the aggregators, but these processes are carried only in BS, that has a high capacity. The authors concluded that this scheme offers energy and time efficiency due to a collection of encryptions and signatures that are sent to the BS. Also, this scheme avoids the signature verification process that takes more time than the signature generation. In addition, it is efficient for large networks. Also, ECDSA algorithm was investigated in detail [16], and contains some of the problems that were related to using inversion processes in generation and verification of the signature. An inversion process consumes a great deal of time, which affects the calculation. This process consumes 10 times the multiplication process. In this scheme, the inversion process was removed, and the results demonstrated that this scheme is more efficient than the original ECDSA scheme. It had less time and therefore less computation and energy. This scheme was applied to the sensor node Micaz.

Using a few keys saved energy, memory and reduced the computation in ECDSA algorithm [87]. Researchers used ECDSA to secure the connection between the gateway and the cluster head(CH), and CH and nodes. This scheme broadcasts only session keys (km), which reduces energy, as the session keys are deleted after their account. This operation leads to less memory (only a few keys stored in memory). The public key of the gateway does not require power, and calculation processes are accomplished through the use of a random number and hash function that require less computation operations. This scheme used periodic authentication in session key. Therefore, it prevents attacks and is appropriate for environments of constrained-source. Finally, modified ECDSA algorithm is proposed through using two curve points publicly (public key Q and point of signature verification (x, y)) [88], where the base point G becomes the private point. In this algorithm, s parameter only is used (without using r parameter) thus removing r overheads. The authors pointed out that the modified ECDSA algorithm is less complex (more performance) using less number of PMs, point addition and point doubling than original ECDSA algorithm.

5. Security Improvement on ECDSA

In this section, we will consider the security in the ECDSA algorithm. A list of mechanisms for improving the security in ECDSA is given. In the beginning, we will come across types of attacks, security requirements and countermeasures by categorizing them into non-physical and physical attacks, as each category includes passive and active attacks. We then will offer security mechanisms to enhance and protect ECDSA's signatures from tampering.

5.1 Categories of Attacks

First of all, we give a review of the attacks that threaten the security level in the ECDSA algorithm. Attacks are becoming increasingly sophisticated, so at the same time there should be countermeasures against such attacks. But these countermeasures are expensive in terms of time, storage, and complex computations. Also, it is difficult to develop a countermeasure for each attack [89].

When the ECDSA algorithm was implemented, it needed to

Table 5. Improvements of different coordinate systems

Paper	Coordinate system	Subsequent improvements of SM (PA and PD)	Coordinate formula	Field
[64]	Affine	1i+2m (PA) and 1i+2m (PD) m is less than Projective	(X,Y)	F_p
[79, 13]	Traditional Projective	12m (PA) and 7m (PD) i is removed 91% is faster than Affine	$(X/Z,Y/Z)$	F_p (192,224,256,384,521)
[80]	LD-Projective	17% (SM) is better than traditional Projective	$(X/Z,Y/Z^2)$	F_{2^m}
[73]	Traditional Projective	13m (PA) and 7m (PD)	$(X/Z,Y/Z)$	F_{2^m} (163,233,283)
	LD-Projective	14m (PA and 4m (PD)	$(X/Z,Y/Z\char`^2)$	
[78]	λ-Projective	LD-Projective: 13m (PA) and 11m (PD) λ-Projective: 11m (PA) and 10m (PD)	$(X/Z,L/Z)$	$F_{2^{254}}$
[81]	LD-Projective with parallel SM	Improved SM execution time	$(X/Z,Y/Z^2)$	$F_{2^{163}}$
[79, 13]	Traditional Jacobian	12m (PA) and 4m (PD)	$(X/Z^2,Y/Z^3)$	F_p (192,224,256,384,521)
[9]	Chudnovsky Jacobian	11m (PA) and 5m (PD)	(X, Y, Z, Z^2, Z^3)	F_{2^m}
[3, 2]	Modified Jacobian	13m (PA) and 4m (PD)	(X,Y,Z,aZ^4)	F_p (160,192,224)
[11]	Affine-Projective	Weierstrass form: 9m (PA) and 1i+2m (PD) Hessian form: 10 (PA) and 1i+5m (PD)	(X,Y) and $(X/Z,Y/Z)$	F_{397} and $F_{2^{163}}$
[24]	Affine-Jacobian	8m (PA) and 4m (PD) 6% is better performance than Jacobian	(X,Y) and $(X/Z^2,Y/Z^3)$	F_{p160}
[9]	Jacobian- modified Jacobiann	11m (PA) and 3m (PD)	$(X/Z^2, X/Z^3)$	F_{2^m}
[19, 83]	Jacobian-Affine and Jacobian	8m (PA) and 4m (PD)	(X,Y) and $(X/Z^2, Y/Z^3)$	F_p (224,256)
[78]	λ-Projective and λ-Affine	8m (PA) and 4m (PD)	$(X/Z,L/Z)$ and $(X,X+Y/X)$	$F_{2^{256}}$

use countermeasures against known attacks [4]. The use of standard criteria to credible organizations such as IEEE, ISO, NIST, NSA, FIPS, and ANSI is extremely important to prevent many attacks. Namely, the abnormal curves are weak for attacks. In addition, the parameters' validation of ECDSA leads to strong security against attacks [21]. Protection of the private key (d) and ephemeral key (k) in the ECC/ECDSA algorithm are essential procedures because if an adversary could get these the keys, the attacker is able to modify messages and signatures and thus the ECDSA algorithm becomes useless. Therefore, a group of countermeasures have used to improve security procedures. Many attacks such as signature manipulation, Bleichenbacher and restart attacks work to retrieve the private key d or ephemeral k generator [21]. A set of conditions (the difficulty of DPL, one-way hash, collision-resistant, k unexpected) are installed to prevent such attacks on the ECDSA. We will classify attacks on the ECDSA algorithm for non-physical and physical attacks. Each category includes many attacks that try to penetrate the signatures of messages in terms of integrity, authentication, and non-repudiation. We will then explain the countermeasures applied by the researchers in their projects.

5.1.1 Non-Physical Attacks and Security Requirements

These attacks do not require physical access (indirect access) to users' devices or the network to penetrate repositories' data or messages transferred between the clients and server. The attackers have used different strategies for physical attacks such as sniffing, spoofing, eavesdropping, and modification to break signatures and encoders transmitted through radio frequency signals or the Internet [90]. They implement non-physical threats to penetrate the security requirements of confidentiality, integrity, and availability (CIA) during the implementation of analysis and modification operations. Non-physical attacks can use network devices indirectly. For example, a DDoS attack uses network devices to send a stream

of messages to disable the server services [91]. These attacks are categorized into passive and active attacks, according to the strategies of these attacks and the intended target of the attack.

5.1.1.1 Passive attacks

These attacks have used eavesdropping, monitoring, and sniffing mechanisms to control and record the activities of network devices, and then break user authentication signatures without altering or destroying the data [92]. The attacker needs to adjust a large number of packets through the use of packet sniffing and packet analysis software to assist in the analysis and penetration of signatures. These attacks are primarily intended to penetrate authentication (signatures), encryption (confidentiality), and access control (authorisation policies). During these attacks, the attacker tries to analyze the data to gain information about messages such as sender and receiver IP addresses, transport protocol such as TCP or UDP, location, data size, and time, as this information helps him in the penetration operation [93]. Passive attacks do not perform any changes to the network data, but the attacker can use these data analyses to destroy the network in the future. This type of attack is difficult to detect by security precautions because the eavesdropper does not perform any data changes. But there is also a set of security requirements that prevent such attacks. Many research projects implement the ECDSA algorithm to prevent the passive non-physical attack. Table 8 shows passive non-physical attacks types in many applications with ECDSA. A brief explanation of some of the most popular passive attacks as follows.

- Traffic analysis and scanning (eavesdropping): Monitors and analyses data as they travel over the Internet or wireless network, for example, monitoring data size, number of connections, open ports, visitor identity and vulnerabilities in operating systems [94].

- Keylogger and snooping: Records authentication activities in victim devices during keystrokes monitoring of username/password [95].
- Tracking: Records and traces users' information such as location for network devices and personal information such as name, address, email, and age [96].
- Guessing: The attacker attempts to guess a user's credential by relying on words like dictionary attack or a combination of symbols (probabilities) such as brute force attack to gain access to the network [97].

5.1.1.2　Active attacks

An active attacker performs data penetration of users' identities by creating, forging, modifying, replacing, injecting, destroying, or rerouting messages as these messages move through the network's nodes or the Internet [93]. This type of attack can use sniff attack (passive attack) to collect information and then performs changes or destruction of network data. For instance, a malicious attacker intercepts money transfer messages in e-banking applications and then implements adding and deleting operations to obtain personal gains [95]. An active attacker can send false or fake signals to deceive the network nodes by linking to the fake server and then redirecting nodes' packets to the legitimate server after modifying the data or authentication messages. Security requirements are significantly important, in particular, the authentication and integrity requirement, to prevent many attacks such as man-in-the-middle (MITM) and replay [98]. According to many research projects in [83, 99, 100, 20, 101], ECDSA's signatures is a security solution to prevent many attacks such as modification, spoofing, denial, and cyber. Figure 2 shows the classification of non-physical attacks. Table 8 shows active non-physical attacks types in many applications with ECDSA. A brief explanation of some of the most popular active attacks is given in the following.

- Spoofing: Many types of attacks such as MITM, replay, routing, and hijacking. The attacker intercepts and modifies credential messages to gain access to the network by MITM attack, while in replay attacks, the attacker intercepts messages and resends them later [97]. Also, routing/hijacking changes the packets paths (changing the IP address) by exploiting the vulnerability in the routing algorithms used to find the best path for moving packets [93] such as IP-spoofing and ARP-spoofing.
- Denial: These attacks are categorized as denial of service (DoS) and distributed denial of service (DDoS). In DoS, the attacker uses his own devices to prevent the server from providing services to network members. This attack targets security requirements (CIA), while in DDoS the attacker uses his own devices as well as network devices to quickly destroy the network [96, 93, 98, 92].
- Cyber: An attacker creates websites such as phishing attacks or social pages for Facebook and Twitter, such as social media attacks, to trick the user into believing that these websites are legitimate websites or pages. The user enters his/her confidential information such as username, password and card number in these counterfeit websites to allow this information to be accessed by the hacker [95, 97].
- Modification: This attack changes, delays, rearranges users' data packets after gaining the attacker's access

as a legitimate user or the attacker may be a legitimate member of the network (internal attack). Any change to this data can cause problems for the user such as changing medical or diagnostic reports in e-health applications [96].

5.1.1.3　Security Requirements

The ECDSA algorithm provides three security requirements: authentication, integrity, and non-repudiation. But integrating the ECDSA algorithm with security mechanisms such as encryption and authorisation provides many security requirements (such as confidentiality, authorisation, availability, accountability, forward secrecy, backward secrecy, auditing, scalability, completeness, anonymity, pseudonymity, and freshness) when used in applications like e-health, e-banking, e-commerce, e-vehicular, and e-governance. We will provide a brief explanation of the security requirements that provided by ECDSA as in the following.

- Authentication has been used to authenticate legitimate users or data in the network to prevent anyone else from accessing it. Namely, if the users' identities or data is a trusted source in the network it is accepted, but if an unknown source it is ignored [102]. Many attacks such as brute-force, keylogger, and credential guessing attempt to penetrate the signatures' authentication service.
- Integrity has been used to ensure that the transmitted data has not been tampered with or edited by the adversary [92]. Many attacks such as MITM, replay, hijacking, and phishing attempt to penetrate the signatures' integrity service.
- Non-repudiation has been used to detect the compromised nodes via the sender who cannot deny his message [98]. Many attacks such as repudiation, masquerading, and social media attempt to penetrate the signatures' non-repudiation service.

5.1.2　Physical Attacks and Countermeasures

Physical attacks are passive or active attacks. The attacker applies a passive physical attack to analyse the signatures and breaks the authentication property (gets the private key) to become a legitimate user in the network. On the other hand, the active physical attack attempts to penetrate the integrity or non-repudiation property to change signatures and messages transferring between the network's nodes. Many physical attacks have applied on ECC/ECDSA. These attacks exploit some problems in these algorithms in order to access the private key [4, 43]. These problems include:

- The power consumption
- Electromagnetic radiation
- Computation time
- Errors

5.1.2.1　Passive attacks

This type of attacks do not tamper with or modify the data but analyses the leaked bits of data (such as the scalar's bits). These attacks take advantage of the different running time for operations, power consumption and electromagnetic radiation. The attacker tries to get some bits leaked from k to produce the full value of k. The attacks that exploit the power consumption and electromagnetic radiation to detect k are called side channel attacks (SCAs). SCA attacks consist of a suitable model, power trace, and statistical phase [4]. In
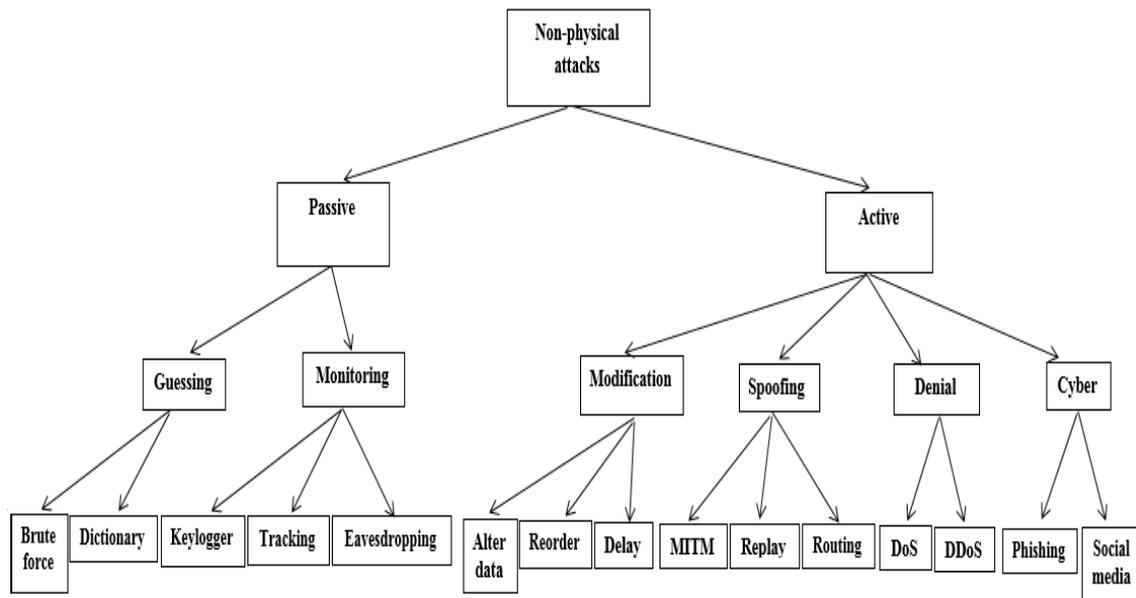
Figure 2. Classification of non-physical attacks

these attacks, the attacker monitors the power consumption and exploits the unintended outputs (side-channel outputs on the secret key) of the device [13]. Leakage power consumption is divided by the transition count leakage and Hamming weight, where the first depends on the state variable bits at a time, while the second depends on the number of 1-bits treated in time [103] by tracking voltages of the device. SCA uses many methods to detect k bits such as the analysis of the distinction between the addition formula operations, creation of a template, statistical analysis, re-use values, special points, auxiliary registers, and the link between the register address and the key [43] (passive attack is described in Figure 3). SCA attacks (passive) consist of four major attacks:

- Simple power analysis (SPA)
  In this attack, the attacker relies on a single trace of power consumption to detect the secret key bits. The attacker extracts these bits based on power consumption discrimination in the addition formula (point addition (PA) and point doubling (PD)) [13, 43].
- Timing attack
  In this attack, the attacker relies on the analysis of the execution time of the addition formula and the arithmetic operations [69]. For example, the attacker analyses the processing time for PA and PD; if the processing time is greater, it is considered PA ("1") or else PD ("0"), and repeats the process until he/she obtains all ephemeral bits k.
- Template attack
  In this attack, the attacker creates templates with a large number of traces of the controlled device. It uses multivariate normal distribution to detect the key based on power consumption during data processing. The attacker gets the ECDSA's key by matching the best template with these traces [43, 104].
- Deferential power analysis (DPA)
  In this attack, the attacker uses many traces in a statistical analysis of power consumption. The attacker uses hypothetical points in SM with stored recorded results. He/she then compares these results with the power consumption of the controlled device to obtain a valid guess

in the detection of secret key bits [105].
General countermeasures have used to prevent passive physical attacks [41]:

- Elimination of relation between data and leakages.
- Elimination of relation between fake data and real data.

The implementation of one of the former two countermeasures ensures data protection from attacks. Jacobi form has been proposed instead of Weierstrass to prevent SCA attacks [106]. This form allows for addition formula operations to have identical time and power and this leads to prevention of SPA and DPA attacks. Unfortunately, this scheme is 70% less efficient than the original scheme (Weierstrass). The average of field operations in their scheme is 3664 while in the original scheme it is 2136. Many recommendations have been presented with PM's endomorphisms to protect many attacks such as Pohlig-Hellman and Pollard's rho through $n \geq 2^{160}$, $\#E(F_q) \neq q$ and n is not dividable on $q^i - 1$ when $1 \leq i \leq 20$ [65]. Similarly, a study presented DSA and ECDSA algorithms in detail and discussed that these algorithms become unsuitable for signing messages (integrity) if applied incorrectly, as this study has focused on the parameters validation of DSA/ECDSA to ensure strong security for these algorithms against different attacks; also, the author proved that these algorithms become strong if the parameters are well-validated [21]. In addition, Dimitrov et al. [56] proposed protection mechanisms with double-base chains method against SCA attack (SPA and DPA), using side channel atomicity against SPA and randomization method against DPA [107]. Also, right-to-left NAF algorithm was proposed without precomputations and investigated security [9]. This algorithm protected against SPA, DPA, and doubling attacks through using atomicity technique countermeasures.
Implementation of template based SPA attack in ECDSA was investigated with microprocessor 32-bit (ARM7 architecture) [104]. The authors pointed out that this attack is applicable on ECDSA through three basics (few bits to retrieve k by lattice attack, using fixed base point and microprocessors enough to build templates) to combine with their attack. Furthermore, Mohamed et al. [69] presented a scheme dependent on combining comb method with width-wNAF. But, this scheme is
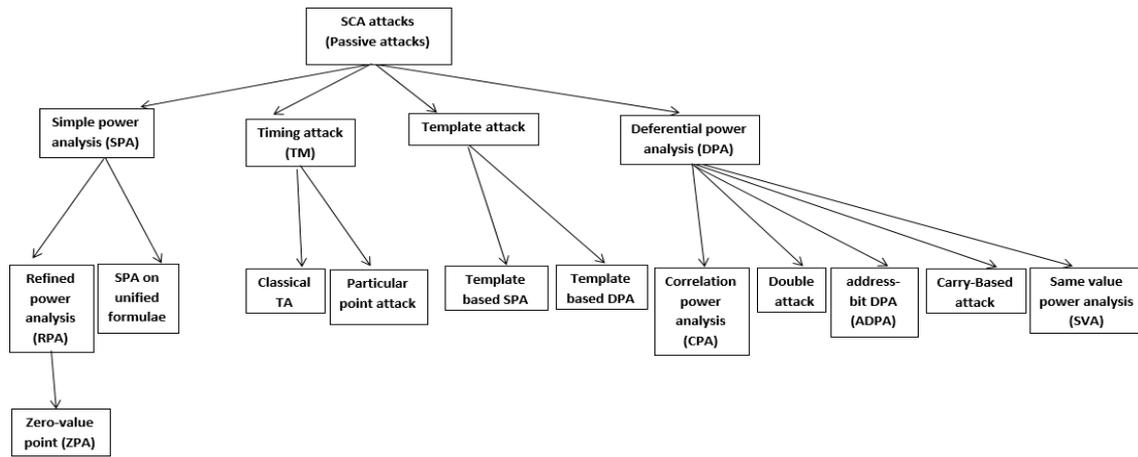
Figure 3. SCA (passive) attacks

not resistant to SCA attacks. Therefore, they suggested using constant runtime to prevent SPA and timing (through add point of infinity (O) when bits value equal nonzero and also add it to precomputation values). Physical attacks are explained on ECC/ECDSA algorithms [43]. The authors described many SCA attacks, as they presented countermeasures and recommendations to apply ECC/ECDSA such as adding randomness, countermeasures selection, and implementation issues to prevent these attacks. Moreover, Rafik and Mohammed [13] analyse SPA and DPA attacks (DPA analysis neglected because it is not viable in the ECC/ECDSA algorithm but SPA is applicable to ECDSA in WSN). DPA is not applicable in the case random scalar but is applicable in the case using it against secret key (d) where the attacker knows signed message [104]. SCA attack is working on the use of information leaked from the SM $(Q = kP)$. They concluded that protecting the ephemeral k is important in ECDSA against SPA attacks.

Template attack with lattice attack is presented to retrieve ECDSA's private key over prime field (F_{p160}) [108]. The authors pointed out that endomorphism curves can be penetrated by Bleichenbacher attack through using one bit of bias. They had a secret key for ECDSA through few number of hours when using 2^{33} signatures, 2^{33} memory and 2^{37} time. Varchola et al. [4] focused on the analysis of correlation power analysis (CPA) attack on ECDSA algorithm in FPGA. They did not use traditional power models (Hamming weight/distance) in the analysis of CPA attack but used another type as a countermeasure (chosen plaintext attack). They proved possible successful feasible CPA in ECDSA on FPGA in case $k^{-1}(e + rd_A)$ or $k^{-1}e + (k^{-1}d_A)r$, but also CPA attack had no impact in ECDSA in case using equation $k^{-1}e + (k^{-1}r)d_A$ (Results obtained by DISIPA platform). Also, a SPA attack was suggested for SM penetration based on conditional subtraction of a modular multiplication. Constant time of modular multiplication is a guarantor to prevent this attack from SM analysis. On the other hand, Liu et al. [109] used secure-SM with randomized point coordinates to prevent SPA, DPA, and ZPA. Table 6 shows the passive physical attacks and countermeasures.

5.1.2.2 Active attacks

There is another type of SCA attack, which uses errors to reveals some bits k, called fault attacks. This type of attack exploits the error in the dummy operations, memory block,

invalid point and curve, and the difference between right and wrong results. The attacker gets an error signature by adding one bit to the generator k. This type of attack is considered a serious risk to ECC/ECDSA algorithms if no proper countermeasures have applied for applications and data transfer environment (Figure 4 shows active SCA attacks). SCA attacks (active) consist of main three attacks:

- Safe-error based analysis
  In this attack, the attacker uses faults to exploit dummy operations (used to resist SPA) and memory blocks (register or memory location). Safe-error includes two types of c safe-error and m safe-error, where the first exploits the vulnerability of the algorithm whereas the second exploits the implementation [41, 110].
- Weak-curve based analysis
  In this attack, the attacker tampers with the curve parameters, where the error is injected by using an invalid point and invalid curve at a specified time for the parameters that were not correctly selected or validated. The attacker uses the errors to move SM from the strong curve to the weak and thus retrieves the private key [111, 43].
- Differential fault analysis (DFA)
  In this attack, the attacker compares the difference between the correct and fault results to retrieve the scalar bits. Differential fault includes two types: classic DFA and sign change FA [43, 112]. Validation of intermediate results and random ephemeral k are used to prevent classic DFA attacks while Montgomery ladder and verify the final results used to prevent sign change FA attacks [41].

General countermeasures have used to prevent active physical attacks [41]:

- Error-detection
- Error-tolerance

The former countermeasure allows the detection of the errors that are added by the attacker whereas the latter prevents the attacker from the discovery of scaler or private key, even after providing faults. The c safe-error attack is described on ECDSA [110]. This attack exploits dummy operations in the addition formula. The attacker can use errors with dummy operations to discover bits of k. In [110] they proposed the use of atomic patterns (Verneuil and Rondepierre patterns) instead of the dummy operations to repeal the c safe-error. Also, Fournaris et al. [112] suggested using a residual number system (RNS) and leak resistant arithmetic (LRA) to prevent both

Table 6. Passive physical attacks and countermeasures

Paper	Passive physical attack (s)	Countermeasure (s)	Year
[106]	SPA/DPA	Jacobi form	2001
[65]	Pohlig-hellman, Pollard's raho Semaev-Satoh-Smart, Weil pairing and Tate pairing	Parameters selection	2001
[21]	Bleichenbacher and restart	Parameters validation	2003
[56] [9] [107]	SPA/DPA/doubling	Side channel atomicity and randomization	2005 2008 2017
[104] [13]	Template/DPA SPA/DPA	Random base point Secure-SM DPA-resistant Random scalar	2008 2013
[69]	SPA/timing	constant runtime	2012
[43]	SPA/DPA/CPA/template RPA/ZPA	Adding randomness appropriate countermeasure selection depend on the application and environments	2012
[108]	Template with lattice	Large prime finite	2014
[4]	CPA	$k^{-1}e + (k^{-1}r)d_A$	2015
[105]	SPA	Constant time modular multiplication, point blinding, random Projective	2015
[109]	SPA/DPA/ZPA	Secure-SM with randomized point coordinates	2017
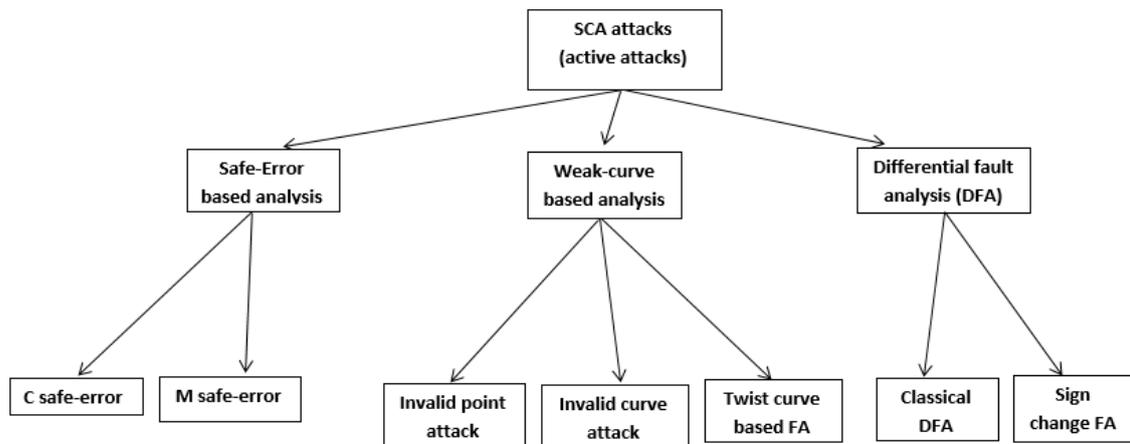
Figure 4. SCA (active) attacks

differential faults (comparing the difference between correct and fault results) and m safe-error (using block memory errors) that are intended to retrieve bits from ephemeral k.

An invalid point attack is carried out on a weak single curve during application skip of faults to move the points to weak Weierstrass curves to retrieve the private key [111]. The authors used point validation as a countermeasure to prevent this attack. Since the device tests the validation of the point on the curve, if it is incorrect, it generates an error message. Moreover, Bos et al. [113] proposed using Edwards curves with ECDSA ($F_{P128,192,256}$) instead of Weierstrass curves to prevent weak-curve attacks. They pointed out that Edwards curves provide constant time and exception-free SM so these attacks are not applicable. Similarly, Harkanson and Kim [6] showed that there were safe and unsafe curves. They demonstrated that curve25519 curve with 256-bit key length is secure and fast.

The comprehensive survey presented information about fault attacks on symmetric and asymmetric cryptography algorithms [89]. The authors presented countermeasures to this attack in different cryptography algorithms. They pointed out that this attack may apply on ECDSA, and also presented countermeasures to prevent this attack on ECDSA. Their countermeasures used against fault attacks are the addition of the cyclic redundancy check (CRC) in the private key or using the public key to verify the signature before sending it, but this process is expensive [89]. In addition, fault

attacks and types are discussed in [43, 114]. The authors have referred to countermeasures to prevent these attacks such as point validation and curve integrity check. According to [115], many countermeasures known to repel various fault attacks such as algorithm restructuring, physical protection of the device, random techniques in computation processes and power consumption with independent implementation. However, the authors noted that fault detection, intrusion detection, algorithmic resistance and correction techniques are capable of eliminating injection fault attacks. Furthermore, Barenghi et al. [116] have provided a fault attack to retrieve the private key. Their attack depended on the injection of errors in the implementation of modular arithmetic operations in ECDSA signatures. They used multiprecision multiplication faults with scalar ($k^{-1}e + (k^{-2}r)kd$) to prevent the penetration of signatures. Recently, Liao et al. [117] discussed invasive SCA attacks such as fault attacks with ECC algorithms (F_{2^m}). The attacker injects errors into the victim's device using tools such as voltage glitch and laser. The authors pointed out that time and location greatly affect the success of fault attacks. Therefore, they proposed one multiplication module and one division module to fix the timing of all operations. Table 7 shows the active physical attacks and countermeasures.

5.2   Security Improvement Via the Random

One of the problems that leads to weak security in the ECDSA algorithm is that there is not sufficiently random and thus ECDSA produces unsafe keys. The random increase in

Table 7. Active physical attacks and countermeasures

Paper	Active physical attack (s)	Countermeasure (s)	Year
[89]	Fault	CRC and public key	2004
[114] [43]	Fault attacks and types	Point validation, curve integrity check, coherence check and combined curve check	2011 2012
[115]	Fault	Fault detection, intrusion, algorithmic resistance and correction techniques	2012
[116]	Invalid point	Point validation	2015
[110]	C safe-error	Atomic pattern	2016
[113]	Weak-curve	Edwards curve	2016
[112]	Differential fault and m safe-error	RNS and LRA (random base permutation)	2016
[116]	Fault	Multiprecision multiplication operations	2016
[117]	invasive fault	Fix computing time	2017

ECDSA prevents penetration of signed messages, whether in physical attacks or during transfer messages from the source to the destination. For instance, Bos et al. [1] pointed out that ECDSA used in Bitcoin suffers from poor randomness in the signature, and this leads to penetration of clients' accounts by attackers. Therefore, the keys should not be duplicated, and k should not be used for more than one message. Also, the randomness source in ECDSA is extremely important to prevent key leakage [118]. If the randomness source is bad this leads to the generation of a bad signature. A bad signature allows the attacker to leak bits of the key and thus discover the private key. Danger et al. [10], Fan et al. [41], Fan and Verbauwhede [43] pointed out to many countermeasures using randomization in scalar (blinding, splitting, and group), point (blinding and coordinates), register address, EC, and field isomorphism. This section explains some researchers' ideas to improve randomness in ECDSA algorithm.

5.2.1 Random of Scalar

Point multiplication (kP) in the ECDSA algorithm uses a random scalar to provide strong signatures as a countermeasure against modifying the data. But the use of scalar k with weak random allows the attacker to reveal k and then forge signatures. The scalar should be subject to several tests to prevent weak random and make it difficult to predict scalar by the attacks. Many methods have used to generate random scalars, such as the use of hash functions or the use of generators.

Randomization source, Hutter et al. [32] used a SHA-1 algorithm to generate ephemeral k and to gain appropriate randomness as a countermeasure against side channel attacks, where the generation of random k is according to FIPS's standard. They pointed out that using a hash function with k increases signature security due to changing k in each message. Also, in the signature algorithm, they used equation $s = k^{-1}e + (k^{-1}r)d$ instead of $s = k^{-1}(e + rd)$ to prevent DPA attacks. Moreover, Nabil et al. [14] used W7 algorithm to increase random integer k in the ECDSA algorithm, which showed that the W7 generator is better than other generators in performance and area. Also, [1] focused on the weaknesses of ECC/ECDSA through the study of four SSH, TLS, Bitcoin and the Austrian citizen card protocols. They discovered that the ECDSA suffers from security weakness in the same way as previous security systems. They focused on the three points: deployment, weak keys, and vulnerable signatures. They gathered databases belonging to the four protocols and found that several cases of repeated public keys have used in SSH and TLS. Bitcoins suffer from several signatures that share nonces that allow an attacker to compute the corresponding private keys and steal coins. The Austrian

citizen card as well suffers from a multi-use of the keys. The big problem with the signature is insufficient randomness in generating keys or nonces in digital signatures. For example, Debian OpenSSL. Koblitz and Menezes [119] discussed random oracle model with ECDSA in the real world. They pointed out this model is less dependent on random generators of weakness and gives more flexibility with modified ECDSA. As they explained, modified ECDSA algorithm prevents chosen-message attacks.

Randomization techniques, randomize with NAF was proposed to prevent DPA attack [120]. This scheme required $n/2$ for a number of point addition and $n + 1$ for point doubling. The authors also proposed using randomized signed-scalar with addition-subtraction multiplication algorithm to prevent SPA attacks. Their algorithm required $n + 1$ for both point addition and doubling. Their scheme is resistant to timing attacks because scalar changed in each running time depending on randomized signed-scalar representation. They explained through experimental results that their scheme is resistant to power attacks. In addition, Samotyja and Lemke-Rust [121] investigated the evaluation of randomization techniques such as scalar blinding and splitting with prime-256, brainpool256r1, and Ed25519 curves in ECC/ECDSA. Their results demonstrated that these techniques prevent SCA attacks. However, Bhattacharya et al. [122] investigated the application of scalar blinding and splitting techniques in ECC/ECDSA with asynchronous samples. They noted that these techniques are vulnerable to branch misprediction, DPA, and template attacks. To eliminate such attacks, they recommended applying execution independence for scalar k or implementing parallel random branching executions. Dou et al. [107] noted that the integration of scalar randomization and side channel atomicity techniques is a countermeasure against SCA attacks such as SPA and DPA. Similarly, Poussier et al. [123] pointed out that scalar blinding is a countermeasure against SPA.

5.2.2 Random of point representation

The random to represent the point prevents an adversary from knowing bits in the point after its representation in binary [103]. Several methods have implemented to represent a base point at random, such as point blinding (add a random point to SM such as $kP + R$), random coordinate systems (such as Projective and Jacobian coordinates) and the use of random isomorphism.

Hedabou et al. [124] used randomness for intermediate values and special points through using randomized linearly-transformed coordinates (RLC) to prevent refined power analysis (RPA) and zero-value point (ZPA) attacks. They pro-

posed using randomized initial point method (RIP) to prevent SPA and DPA attacks where RIP uses random point (R) with point multiplication (kP). ESDSA is vulnerable to fault attack. Schmidt and Medwed [125] presented a fault attack on ECDSA, attacker modifies program flow and tries to benefit from error result to get parts (bits) for ephemeral key k, then performs lattice attack to obtain the private key (d). The authors implemented this attack on point multiplication through Montgomery ladder and D&A algorithms. The success of this attack is probability, as this attack tries to get on bits from k through many signatures. They used the Jacobian coordinate system through implementation because it is efficient. Also, the authors used countermeasure to fault attack through using a discrete algorithm l in point representation $(Q = (X : Y : Z; l)$, where modifying by the adversary for point addition and point doubling is detection. This scheme can prevent fault and SCA attack (with random mask). Using randomness point is recommended to prevent DPA attacks [56, 69]. Liu et al. [126] adopted a randomized point in PM to protect ECDSA signatures. They pointed out that a randomized point application with double-and-add-always prevents SM's bits from being attacked against multiple attacks such as ZPA, SPA, and DPA. Also, point randomization with curve Curve448 in ECC/ECDSA applied to transport layer security (TLS) [127]. The results of this research indicate that this protocol offers a 224-bit security level better than applying the Curve25519 curve with 127-bit randomization technology against SCA attacks.

Furthermore, Medwed and Oswald [104] described countermeasure to prevent template based SPA attack using randomness coordinates in DPA resistance. Also, using verification batches reduces the total time of ECDSA verification but these batches suffer from some attacks. Randomizers used symbolic computation in ECDSA batch verification to improve security level against attacks [128]. The authors used D&A and Montgomery ladder algorithms in their scheme. They noted that using numeric computation with randomizers in x-coordinate (D&A) is faster and more secure than using Montgomery ladder algorithm in their scheme but original D&A is faster than their scheme. Countermeasures (randomness coordinate) were used to prevent implementation attacks [129, 124, 32, 33, 130]. A randomized Projective coordinate (RPC) was used as a countermeasure to prevent differential power analysis (DPA) and template based SPA attacks. RPC has implemented through a random application in the Z coordinate to represent the intermediate points in PM.

5.3 Security Improvement Via PM Methods

In Section 4, we have discussed how the PM method is used to improve ECDSA efficiency. In this Subsection, we will consider how the PM method is used to improve security in ECDSA. Improvement of the efficiency of the PM's methods is dramatically important for ECDSA algorithm, particularly when used in restricted-source environments, but at the same time maintaining the security of this algorithm is more important because the basic mission of this algorithm is security through integration of messages and protection from the modification, as well as authentication of messages and users in the network. Many attacks on the security of this algorithm have applied. Therefore, many ideas are presented by researchers to improve the security of PM's methods. PM's method such as D&A, Window, NAF, Comb, and Montgomery is proposed to improve kP through reducing the number of operations of addition and doubling. The attackers are taking advantage of these operations (such as the computation time in addition operation is longer than doubling operation) to obtain a secret key or ephemeral key. Therefore, this Section describes some of the recently improved methods for PM's security.

Liardet and Smart [106] used combining point blinding and randomized signed window method as a countermeasure against SPA and DPA attacks. Also, an improvement on left-to-right Window method was proposed with standard curves such as NIST and SECG as a countermeasure against SCA attacks [131]. The author added a special signed digit (-2^w) to $\{1, 2, ..., 2^w - 1\}$ in scalar representation in order to get addition formula operations in the uniform formula instead of using dummy addition that were revealed by attackers, thus preventing SCA attacks. In addition, Möller [132] improved on the idea in [131] through using right-to-left Window method with 2^w-ary. This scheme avoided using fixed table (used in [131]) through randomization Projective, where this table allows attackers to analyze information statistically. They used two-processor in their scheme to improve the performance of PM through parallel. Moreover, window and Montgomery algorithms were improved to prevent SPA and DPA attacks [8]. The small multiple was added to ephemeral key with the idea of [131] in window method and combining addition and doubling operations into the uniform formula in Montgomery method where original Montgomery compute separately addition and doubling operations. Improvement on fixed-base comb method was presented against SCA attacks [124]. Furthermore, the authors used sequence non-zero of bit-strings (i.e. removing all zeros only using 1 and -1) to represent the ephemeral key. They pointed out that their scheme costs a little more computation time than the original comb method.

Medwed and Oswald [104] used a double-and-add-always algorithm to prevent SPA attacks in their scheme. Hutter et al. [32] used Montgomery ladder algorithm to improve PM's performance and countermeasure against SPA attacks in their scheme. Rafik and Mohammed [13] carried out SPA on many methods for PM such as double-and-add, window, and comb. They found comb method is optimal for their application to prevent SPA attacks. Their results explained that the Secure-CM was both more efficient timely (1.57s) and secure in its implementations. Pessl and Hutter [33] used countermeasures to prevent implementation attacks. They used zeroless signed digit (ZSD) with comb method to prevent simple power analysis (SPA).

Karaklajić et al. [133] merged the López-Dahab method with the Montgomery ladder. They pointed out that their scheme provides security against FA attacks by using the Montgomery method for coherent check and high probability detecting injected errors. Oliveira et al. [134] have shown that the Montgomery ladder with Galbraith-Lin-Scott (GLS) curves offers a 128-bit security level against SCA attacks. In 2017, Feng and Li [135] have noted that the ECC/ECDSA (256-bit) Comb method has a bit leakage by a SPA attack. They improved the Comb method by adding a signed representation to ephemeral k to prevent SPA attacks completely.

6. ECDSA Applications

Recently, many applications have become available that store and move sensitive data for users between network nodes and server. These applications require powerful encryption and signature algorithms (such as ECC/ECDSA) to protect users' information. We will show the most important applications that use the ECDSA algorithm to provide security and privacy requirements for this data. Table 8 shows ECDSA and recent applications with non-physical attacks.

6.1 Electronic-Health

E-health provides services that allow healthcare providers and patients to share medical records across various health centers, such as hospitals, clinics, and even the home. These services provide facilities to improve the health of patients. Because of efficient methods of electronically sharing patient data rather than traditional paper-based methods, patient health data is available anywhere and anytime for healthcare providers and patients. Health institutions and researchers are seeking to develop these applications to improve the quality of care, disease diagnosis, and remote medical surveillance [138]. E-health includes many systems such as electronic health record (EHR), electronic medical record (EMR) and personal health record (PHR), which are used efficiently to share medical records either globally (EHR) or locally (EMR) and are administered either by the authority provider (EHR and EMR) or by the patient (PHR) [139]. These applications also suffer from many problems such as complexity in computations, lack of resources, precision management, and mobility. However, the main problem that threatens the acceptance of these systems for patients and providers is the privacy of patients' data. This data or medical reports from WSN, laptop, and phone to an e-health server are vulnerable to attacks and intrusion because the Internet and wireless network are unsafe environments.

Data privacy is a key issue for any e-application specifically for e-health applications. These applications require security and privacy mechanisms such as authorisation policies, encryption algorithms, and robust signatures to protect medical repositories for patients from malicious attacks. Many examples of security refer to threats implemented against e-health:

- In 2013, there were penetration attacks on healthcare data in US hospitals. These attacks revealed 85.4% of the medical records (protected health information (PHI)) of the 5 largest incidents for patients' data [140].
- In 2016, Apple Health (Medicaid) was exposed for data breach. This attack revealed 370,000 records for clients in Apple Health (Washington state) [141].
- In 2017, an unauthorised individual penetrated EHR the New Jersey Diamond Institute for Fertility and Menopause. The hacker revealed PHI to 14633 record containing patients' information such as names, birth dates, social security numbers, and sonograms [142].

According to the Vormetic report on data security 2016, healthcare is one of the sectors that is most vulnerable to hackers' attacks and thus requires increased efforts to secure health data by 64%. This report stated on 21 January 2016 that 91% of enterprises suffer from vulnerability threatening data security (internal and external attacks). The study of security data included several countries such as Australia, USA and Germany [143]. Therefore, many systems such as national e-health transition authority (NEHTA) in Australia and health insurance portability and accountability act (HIPAA) in the USA recommended applying security and privacy in e-health applications correctly and accurately to prevent security threats [144].

In addition, the accuracy of medical data is critical in diagnosing diseases and determining the condition of patients during their online or wireless transmission from patient to a healthcare provider. Also, penetration of the medical data with diseases such as addiction, HIV infection, sexual and dermatological conditions can lead to harassment, discrimination, even death of the patient if data reports change during the transfer from client to server [145]. These penetrations show that medical records in e-health require a high level of security and privacy. Many modern projects implement the ECDSA algorithm for high security and efficient performance. ECDSA signatures (Koblitz with the key of 163-bit) and energy-harvesting techniques were proposed to monitor patients' conditions by WSN in e-health [146]. The authors applied the patient's health surveillance scenario during the deployment of a set of sensor nodes such as TelosB, Micaz, and MagoNode ++. They generated a signature with a time of 300ms and an energy consumption of 7mj. They concluded that these results reduce time and energy in WSN to serve e-health applications that deal with massive data. Masud and Hossain [99] proposed using ECC/ECDSA to protect patients' information in collaborating clouds healthcare based on pairing based cryptography. Also, Ibrahim and Dalkılıç [96] designed a scheme to support the privacy and security of patients' data in e-health. This scheme applied the ECDSA's signatures in the RFID tag to provide mutual authentication in the protection of medical records. They pointed out that their scheme is capable of preventing man-in-the-middle attacks, tracking, eavesdropping, impersonation, replay, and cloning. They used the Shamir scheme to reduce the cost of PM in signature verification (ECDSA) as well as security support. Their scheme provided fewer calculations and more security in the process of authentication of medical records in e-health.

6.2 Electronic-Banking

E-banks have used the personal information of their customers to authorise access to their bank accounts. Many systems have applied for dealing with bank accounts such as online banking (OB), mobile banking (MB), credit card (CC) [147] and automated teller machines (ATM) which require signature algorithms to protect confidential information for customers. E-banking implements a set of security measures to prevent hackers and Internet thieves. Hackers are trying to create a gap in these systems to hack users' accounts. They have applied high-tech to penetrate customers' credits. Therefore, the process of authentication for legitimate users in e-banking applications is extremely important. Many examples of security refer to threats implemented against e-banking applications:

- In 2012, millions of clients blocked access to their accounts at the Royal Bank of Scotland [148].
- In 2015, the website www.000webhost.com was hacked by a malicious attack. The result was the attacker's access to 13 million user accounts; the hacked data contained personal information for customers (such as name and plaintext password) [95].
- In 2016, DDOS attacks were perpetrated against HSBC bank, which is one of the biggest banking names in the

Table 8. ECDSA and applications against passive and active non-physical attacks

Paper and year	Passive non-physical attack (s)	Active non-physical attack (s)	Signature algorithm and key size	Application
[99] - 2017	-	Masquerade, MITM, replay and target oriented	ECC/ECDSA (160-512 bit) with pairing cryptography	E-health (EHR, EMR, PHR)
[96]- 2017	Tracking and eavesdropping	Impersonation, replay, MITM and cloning	ECDSA, ECDH, AES and Shamir's trick	
[95] - 2017	Shoulder surfing, keyloggers password-related and direct observation	Phishing and data breach incidents	ECDSA (256-bit) and SHA-256	E-banking (OB, MB, CC, ATM)
[97] - 2017	Wifi sniffing	Phishing, impersonation, MITM and reflection	ECDSA (224-bit) and ECIES	
[83] - 2017	Stealthy	Heartbleed	ECDSA (256-bit) to support SIGaaS	E-commerce (Bitcoin,Litcoin, Freicoin, Peercoin)
[101] - 2017	Eavesdropping	MITM	ECDSA (256-bit) with threshold signatures	
[20] - 2017	-	Fake entity and position counterfeit	ECDSA (160,192,224, 256, 384, 521) with various curves	E-vehicular (V2V, V2I)
[98] - 2017	Tracking	Modification, Sybil, impersonation, DoS, bogus information. and replay	ECDSA (224-bit) with pseudonym	
[136] - 2015 [102] - 2015	-	Private key recovery and tampering	Signatures algorithms (ECC/ECDSA, RSA) with one time public key infrastructure and parameters selection	E-governance (G2C, G2B, G2E,G2G)
[137] - 2013 [100] - 2016	-	Cyber	ECC/ECDSA	

world; these attacks led to the suspension of service for two days and prevented customers from accessing their accounts [148]. In the current era, authentication systems based on username/password are no longer safe in protecting user accounts [95]. We have investigated the adoption of ECDSA signatures in modern e-banking applications and their ability to prevent hacking threats. In 2017, Alhothaily et al. [95] proposed a scheme to protect e-banking applications. They designed a protocol to authenticate customers in e-banking applications. Also, they pointed out that their scheme prevents e-banking attacks such as phishing, shoulder surfing, keyloggers, data break accidents and password-related attacks. Their scheme relied on the ECDSA (256-bit) algorithm with SHA-256 to sign the ticket. This ticket implements security mechanisms to protect users' accounts such as one-time username, session key, ticket validity period, timestamp and add account permission for each login operation. They pointed out that their scheme provides protection against previous attacks as it prevents reuse of credits. Moreover, Bojjagani and Sastry [97] discussed that short message service (SMS) is not reliable in e-banking applications. These applications exchange these messages with their customers in a dense and daily way over the Internet or wireless network. They proposed a model for protecting bank messages by using ECDSA (p-224) and the elliptic curve integrated encryption scheme (ECIES). They integrated the ECIES algorithm with signature verification in ECDSA to support the authentication process in e-banking applications. Their model achieves three stages (authentication, transaction, and notification) to complete the authorisation process between sender and receiver. They pointed out that their scheme is efficient and reliable for real-time banking applications, against the attacks of wifi sniffing, reflection, replay, man-in-the-middle, phishing, and impersonation, and provides end-to-end communication security between the client and server. They demonstrated that the ECC/ECDSA algorithm provides high performance in e-banking applications because these algorithms use small keys and an appropri-

ate computation cost for these applications compared to other public key algorithms (such as RSA, DSA, and DH).

6.3 Electronic-Commerce

The Internet is important medium for providing services, sharing information, buying and selling electronic products, applications, business transactions on an internal or external e-commerce level. Business and e-commerce transactions information have been shared between several parties, such as governments, companies, and customers (consumers) via websites. The Internet has brought a tremendous amount of positives but it has also made some negatives. The most important drawback is hacking threats to personal and sensitive information to get gains and benefits [83]. The progress of the huge development of this technology has brought with it security risks that threaten the personal information of users and attempt to detect private keys in signing and encrypting data. Due to a high demand for consumers and companies to market products online, e-commerce requires secure signature algorithms and is efficient to deal with tens of thousands of requests per second. Many examples of security refer to threats implemented against e-commerce:

- In 2016, the report in [149] pointed out that many popular websites have been hacked by credential replay attacks (password reuse) for more than three billion user accounts. Fraud attacks on e-commerce in the united states are expected to hit $7 billion by 2020.
- In 2016, phishing attacks target e-commerce where 44% of threaten customers and 11% of public activity [150].
- In 2017, the study of IP converge data services (IPC) stated that more than 16,600 DDoS attacks were revealed on e-commerce deals particularly on Valentine's Day and Chinese New Year [151].

Many currency signing systems have used in e-commerce applications such as Bitcoin, Litecoin, Freicoin and Peercoin [152]. Most of these systems implement ECDSA signatures to support security features (authentication, integrity, and non-repudiation). A scheme to secure e-commerce and

online transactions was proposed in [83]. The authors developed a graphic processing unit-universal elliptic curve signature server (Guess) with the adoption of the ECDSA signature mechanism (key of 256-bit binary) to provide signatures with high-performance computations. They applied signature as a service (SIGaaS) on an enterprise private cloud (EPC) to secure e-payment over the Internet. They pointed out that the Internet is an insecure medium because of the variety of attacks that try to hack private keys. Also, they described SIGaaS as impervious to private key attacks such as heartbleed and stealthy newly emerged. Their scheme described the authentication issues between tenants and Guess. Therefore, they have demonstrated that e-commerce applications require robust authentication. For instance, Chinese Alipay website fulfills huge requests and exchanges personal information for users. This website performs more than 85,900 transactions per second. Therefore, this website requires an efficient signature algorithm in terms of speed, time and security requirements to protect these transactions online. The authors stated that the graphics processing unit (GPU) with ECDSA showed high performance and security in dealing with e-commerce websites such as the games market. Furthermore, Mann and Loebenberger [101] used ECDSA to implement two-factor authentication with e-commerce applications (Bitcoin wallet). They implemented a mutual signature protocol with signature verification over a separate communication channel. Their scheme applied a threshold to support a set of signatures, instead of one signature as well as using the hash value in checking the payer and payee. They argued that ECDSA and threshold signatures solved Bitcoin security problems such as monitoring transaction details (block chain public) and transaction size. They also pointed out that randomization is significantly important in preventing malicious attackers from discovering the secret key in ECDSA.

6.4 Electronic-Vehicular

Vehicular ad hoc network (VANET) applications are important modern applications to secure people's lives on the road. These applications are a collection of network cars that share information such as police, emergency, and smart taxi cars. These applications regulate network traffic to ensure safety for users while enforcing road laws. Several features for VANET applications have used such as self-regulation, distributed communications environment, and dynamic topology. These applications have been designed to avoid traffic congestion, prevent accidents, and prevent violations of road rules. The primary objective is to protect passengers on the road. This type of network operates with two categories: vehicle to vehicle (V2V), and vehicle to infrastructure (V2I). VANET is a class of mobile ad hoc network (MANET) network but it works regularly and not randomly. The critical issue and great challenge for this type of application is security due to the VANET network sharing nodes information in the open medium [153]. To protect users in a VANET network, encryption and signature mechanisms should be used to prevent the risks of modifying messages between nodes. Many attacks threat the VANET application such as impersonation, Sybil, modification, DDoS and reply. To prevent these attacks the confidentiality, integrity, and availability (CIA) security requirements should be applied. Many examples of security refer to threats implemented against e-vehicular:

- In 2015, a hacker successfully carried out a cyber attack to hijack and control a Jeep car over the Internet. This attack prevented the driver from taking control of his car [154].
- In 2017, the report in [155] stated that a hacker penetrated board messages for highways in California. This hacker targeted to distract drivers and endanger them. The report reported that 58.85% of attacks such as DDoS, cyber and brute forcing were rated as high risk for VANET applications.

The ECDSA algorithm was used to authenticate messages with VANET [156]. Though the results, the authors noted delay generation key, signature, and verification increase with an increase in curve size. This scheme provides integrity, authentication, and non-repudiation, as it is convenient for constrained-source networks. Also, Franeková et al. [20] have provided a security analysis of the ECDSA algorithm (key lengths of 160, 192, 224, 256, 384 and 521) with VANET (V2V) applications. They applied many curves to the ECDSA (OpenSSL library) such as prime, pseudorandom binary, Koblitz binary. They also focused on data transfer scenarios and used the Riverbed Modeler tool to analyze data transferred between vehicles. Their scheme discussed whether VANET can be exposed to many attacks such as fake entity, position counterfeit, and Sybil. To generate sufficient security in these applications, at least the NIST p-224 and NIST p-256 keys should be used in ECDSA. They noted that the important parameters that affect the efficiency of signatures in an e-vehicular environment are the size and time of signature generation and verification. They concluded that ECDSA p-224 or p-256 provides efficiency (time and size of data) and adequate security to secure e-vehicular (V2V) applications. In addition, Rajput et al. [98] argued that the privacy of location information and identity should be safe in e-vehicular applications because changing this data could lead to fatal accidents. They applied a method of authentication based mainly on the signatures of ECDSA with a pseudonym to provide anonymity to the vehicular applications. Also, they pointed out that their scheme is capable of preventing malicious attacks such as location tracking, modification, impersonation, bogus information, Sybil, replay, and DOS. Their scheme supports standard security requirements as well as traceability and low overheads. In their scheme, the pseudonym consists of signing ECDSA, timestamp and a random number. They pointed out that a lightweight pseudonym is sufficient to support security in e-vehicular applications against malicious attacks.

6.5 Electronic-Governance

E-governance is an application that provides services to stakeholders through various communication technologies such as Internet, SMS, and email to simplify the procedures of services for citizens, business, and government agencies. This application consists of four types of transactions involving government to citizens (G2C), government to business (G2B), government to employee (G2E), and government to government (G2G) [157] to deal with various data types of stakeholders. The adoption of these applications in society should meet a high level of citizen confidence through a range of requirements such as flexibility, transparency, accountability, manageability, good governance, as well as trustwor-

thy (security and privacy) services [158]. According to a United Nations report, a significant increase is in the development of e-governance applications in some countries of the world, but the issue that restricts their use is the security of data and the privacy of the stakeholders (citizen, government, business, employees, intermediaries and service providers) against hacking attacks [159]. These applications are a large pool of databases of diverse sources; therefore, these applications are exposed many risks such as legitimate users, e-government employees, foreign intelligence agencies, business enterprises, terrorist organizations, software providers, and natural disasters. Sharma and Kalra [157] referred to several attacks against e-governance such as DOS, insider, impersonation, replay, and offline password guessing. Many examples of security refer to threats implemented against e-governance:

- In 2012, 112 Indian government websites such as the Planning Commission and the Finance Ministry were hacked. The hacker stopped these websites from working for weeks [160].
- In 2017, a cyber attack was carried out against Australian government websites such as the Finance Department and Australian Electoral Commission. This attack revealed online sensitive information of more than 50,000 records [161].

To secure citizen's data in e-governance applications, security requirements should be applied. Several procedures have used to secure the privacy of users such as server security, network security, data security, workstation security as well as physical and environmental security [159]. Most developed countries such as USA, UK, and some other NATO member countries use ECC/ECDSA to secure the privacy of citizens' data, as well as to protect interoperability between governments [162] and internal communications to prevent data fraud attacks. Jain et al. [136] have demonstrated the importance of applying public signatures to e-governance that authenticate stakeholders in the system. The availability of efficient digital signatures will provide a powerful e-authentication mechanism that will allow citizens to safely handle government documents (such as property tax, income tax, license, visa, passport, and identity card services). These applications invest technologies such as cloud computing and Big Data in dealing with large amounts of citizen data, but also provide significant challenges for developers of these technologies in terms of security and privacy. Moreover, Singh et al. [102] argued that e-governance applications store large amounts of citizens' data in various technologies such as cloud computing. They proposed applying ECDSA signatures to protect cloud computing data from change with recommendations for using ECDSA with accurate parameters. In addition, Lo'ai et al. [100], Zefferer et al. [137] have noted that ECC/ECDSA is the best algorithm for signing and encrypting sensitive data transmitted through e-governance applications. They also discussed that several SCA attacks try to penetrate the private key in ECDSA. Furthermore, they explained that there is no countermeasure to block all hacker attacks, but a set of countermeasures should be used to secure data sharing.

7. Conclusion and Future Work

In this paper, we made a detailed study of the security and efficiency of the ECDSA algorithm. This study enabled us to see the recent improvements for ECDSA in terms of security, efficiency, and application. This updated improvement may possibly inspire some ideas to improve some of the methods in the security and efficiency of the ECDSA. We hope that this study will be useful for researchers to locate research ideas in securing and improving the efficiency of the ECDSA algorithm, especially when used in the constrained source devices. Several future works that we intend to accomplish:

- We intend to implement the ECDSA algorithm in e-health applications to apply the integrity and authentication properties to protect patients' data by the design of the authentication and authorisation protocols for users in the network.
- We intend to implement lightweight curves such as Edward curve, the Montgomery SM method, and λ-Projective in [78] that are critical procedures to increase the efficiency of implementing ECDSA signatures in accessing a massive database of e-health.
- We intend to use lightweight hash algorithms to generate random ephemeral k and personal information for healthcare users. These algorithms provide high performance in computations operations.
- We intend to investigate the application of the ECDSA algorithm with homomorphic and anonymity mechanisms to secure source-constrained devices data such as WSN. These devices require efficient and secure mechanisms to protect patients' data when transferred from WSN to e-health.

References

[1] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 157–175.

[2] M. Aydos, T. Yanik, and C. Koc, "High-speed implementation of an ecc-based wireless authentication protocol on an arm microprocessor," *IEE Proceedings-Communications*, vol. 148, no. 5, pp. 273–279, 2001.

[3] H. Cohen, A. Miyaji, and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 1998, pp. 51–65.

[4] 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 (ReConFig)*. IEEE, 2015, pp. 1–6.

[5] E. Barker and Q. Dang, "Nist special publication 800–57 part 1, revision 4," 2016.

[6] R. Harkanson and Y. Kim, "Applications of elliptic curve cryptography: a light introduction to elliptic curves and a survey of their applications," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*. ACM, 2017, p. 6.

[7] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[8] T. Izu, B. Möller, and T. Takagi, "Improved elliptic curve multiplication methods resistant against side channel attacks," in *International Conference on Cryptology in India*. Springer, 2002, pp. 296–313.

[9] M. Joye, "Fast point multiplication on elliptic curves without precomputation," in *International Workshop on the Arithmetic of Finite Fields*. Springer, 2008, pp. 36–46.

[10] J.-L. Danger, S. Guilley, P. Hoogvorst, C. Murdica, and D. Naccache, "A synthesis of side-channel attacks on elliptic curve cryptography in smart-cards," *Journal of Cryptographic Engineering*, vol. 3, no. 4, pp. 241–265, 2013.

[11] N. P. Smart and E. Westwood, "Point multiplication on ordinary elliptic curves over fields of characteristic three," *Applicable Algebra in Engineering, Communication and Computing*, vol. 13, no. 6, pp. 485–497, 2003.

[12] J.-H. Han, Y.-J. Kim, S.-I. Jun, K.-I. Chung, and C.-H. Seo, "Implementation of ecc/ecdsa cryptography algorithms based on java card," in *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*. IEEE, 2002, pp. 272–276.

[13] M. B. O. Rafik and F. Mohammed, "The impact of ecc's scalar multiplication on wireless sensor networks," in *Programming and Systems (ISPS), 2013 11th International Symposium on*. IEEE, 2013, pp. 17–23.

[14] G. Nabil, K. Naziha, F. Lamia, and K. Lotfi, "Hardware implementation of elliptic curve digital signature algorithm (ecdsa) on koblitz curves," in *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*. IEEE, 2012, pp. 1–6.

[15] K. Eisenträger, K. Lauter, and P. L. Montgomery, "Fast elliptic curve arithmetic and improved weil pairing evaluation," in *Cryptographers' Track at the RSA Conference*. Springer, 2003, pp. 343–354.

[16] H. Zhong, R. Zhao, J. Cui, X. Jiang, and J. Gao, "An improved ecdsa scheme for wireless sensor network," *International Journal of Future Generation Communication and Networking*, vol. 9, no. 2, pp. 73–82, 2016.

[17] B. Driessen, A. Poschmann, and C. Paar, "Comparison of innovative signature algorithms for wsns," in *Proceedings of the first ACM conference on Wireless network security*. ACM, 2008, pp. 30–35.

[18] T. Cheneau, A. Boudguiga, and M. Laurent, "Significantly improved performances of the cryptographically generated addresses thanks to ecc and gpgpu," *computers & security*, vol. 29, no. 4, pp. 419–431, 2010.

[19] S. Xu, C. Li, F. Li, and S. Zhang, "An improved sliding window algorithm for ecc multiplication," in *World Automation Congress (WAC), 2012*. IEEE, 2012, pp. 335–338.

[20] M. Franeková, P. Holečko, E. Bubeníková, and A. Kanáliková, "Transport scenarios analysis within c2c communications focusing on security aspects," in *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*. IEEE, 2017, pp. 000 461–000 466.

[21] S. Vaudenay, "The security of dsa and ecdsa," in *International Workshop on Public Key Cryptography*. Springer, 2003, pp. 309–323.

[22] W. Osamy and A. M. Khedr, "An algorithm for enhancing coverage and network lifetime in cluster-based wireless sensor networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 10, no. 1, 2018.

[23] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Third IEEE international conference on pervasive computing and communications*. IEEE, 2005, pp. 324–328.

[24] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on mote sensors (short paper)," in *International Conference on Information and Communications Security*. Springer, 2006, pp. 519–528.

[25] P. Trakadas, T. Zahariadis, H. Leligou, S. Voliotis, and K. Papadopoulos, "Analyzing energy and time overhead of security mechanisms in wireless sensor networks," in *2008 15th International Conference on Systems, Signals and Image Processing*. IEEE, 2008, pp. 137–140.

[26] G. De Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2008, pp. 580–585.

[27] X. Fan and G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 723–736, 2012.

[28] R. K. Kodali, "Implementation of ecdsa in wsn," in *Control Communication and Computing (ICCC), 2013 International Conference on*. IEEE, 2013, pp. 310–314.

[29] M. Lavanya and V. Natarajan, "Lightweight key agreement protocol for iot based on ikev2," *Computers & Electrical Engineering*, 2017.

[30] M. LAVANYA and V. NATARAJAN, "Lwdsa: light-weight digital signature algorithm for wireless sensor networks," *Sādhanā*, pp. 1–15, 2017.

[31] T. Kern and M. Feldhofer, "Low-resource ecdsa implementation for passive rfid tags," in *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 1236–1239.

[32] M. Hutter, M. Feldhofer, and T. Plos, "An ecdsa processor for rfid authentication," in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, 2010, pp. 189–202.

[33] P. Pessl and M. Hutter, "Curved tags–a low-resource ecdsa implementation tailored for rfid," in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, 2014, pp. 156–172.

[34] M. I. Younis and M. H. Abdulkareem, "Itpmap: An improved three-pass mutual authentication protocol for secure rfid systems," *Wireless Personal Communications*, vol. 96, no. 1, pp. 65–101, 2017.

[35] A. Ibrahim and G. Dalkılıç, "An advanced encryption standard powered mutual authentication protocol based on elliptic curve cryptography for rfid, proven on wisp," *Journal of Sensors*, vol. 2017, 2017.

[36] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "Iot applications on secure smart shopping system," *IEEE Internet of Things Journal*, 2017.

[37] L. Batina, J.-H. Hoepman, B. Jacobs, W. Mostowski, and P. Vullers, "Developing efficient blinded attribute certificates on smart cards via pairings," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2010, pp. 209–222.

[38] M. Savari and M. Montazerolzohour, "All about encryption in smart card," in *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*. IEEE, 2012, pp. 54–59.

[39] J. Dubeuf, D. Hely, and V. Beroulle, "Enhanced elliptic curve scalar multiplication secure against side channel attacks and safe errors," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2017, pp. 65–82.

[40] G. M. de Dormale and J.-J. Quisquater, "High-speed hardware implementations of elliptic curve cryptography: A survey," *Journal of systems architecture*, vol. 53, no. 2, pp. 72–84, 2007.

[41] J. Fan, X. Guo, E. De Mulder, P. Schaumont, B. Preneel, and I. Verbauwhede, "State-of-the-art of secure ecc implementations: a survey on known side-channel attacks and countermeasures," in *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 76–87.

[42] A. S. Abdouli, J. Baek, and C. Y. Yeun, "Survey on computationally hard problems and their applications to cryptography," in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*. IEEE, 2011, pp. 46–52.

[43] J. Fan and I. Verbauwhede, "An updated survey on secure ecc implementations: Attacks, countermeasures and cost," in *Cryptography and Security: From Theory to Applications*. Springer, 2012, pp. 265–282.

[44] Q. Yang, X. Zhu, H. Fu, and X. Che, "Survey of security technologies on wireless sensor networks," *Journal of Sensors*, vol. 2015, 2015.

[45] H. Mayer, "Ecdsa security in bitcoin and ethereum: a research survey," 2016.

[46] H. Li, R. Zhang, J. Yi, and H. Lv, "A novel algorithm for scalar multiplication in ecdsa," in *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on*. IEEE, 2013, pp. 943–946.

[47] M. Hutter, M. Feldhofer, and J. Wolkerstorfer, "A cryptographic processor for low-resource devices: Canning ecdsa and aes like sardines," in *IFIP International Workshop on Information Security Theory and Practices*. Springer, 2011, pp. 144–159.

[48] A. M. Ismail, M. R. M. Said, K. M. Atan, and I. S. Rakhimov, "An algorithm to enhance elliptic curves scalar multiplication combinig mbnr with point halving," *Applied Mathematical sciences*, vol. 4, no. 1, pp. 259–1, 2010.

[49] F. Morain and J. Olivos, "Speeding up the computations on an elliptic curve using addition-subtraction chains," *Information Theory Applications*, vol. 24, no. 6, pp. 531–543, 1990.

[50] G. Purohit and A. S. Rawat, "Fast scalar multiplication in ecc using the multi base number system," *International Journal of Computer Science Issues*, vol. 8, no. 1, pp. 131–137, 2011.

[51] G. Purohit, A. S. Rawat, and M. Kumar, "Elliptic curve point multiplication using mbnr and point halving," *Int. J. Advanced Networking and Applications*, vol. 3, no. 2, pp. 1329–1337, 2012.

[52] K.-W. Wong, E. C. Lee, L. Cheng, and X. Liao, "Fast elliptic scalar multiplication using new double-base chain and point halving," *Applied mathematics and computation*, vol. 183, no. 2, pp. 1000–1007, 2006.

[53] E. W. Knudsen, "Elliptic scalar multiplication using point halving," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 1999, pp. 135–149.

[54] M. Ciet, M. Joye, K. Lauter, and P. L. Montgomery, "Trading inversions for multiplications in elliptic curve cryptography," *Designs, codes and cryptography*, vol. 39, no. 2, pp. 189–206, 2006.

[55] M. Ciet and F. Sica, "An analysis of double base number systems and a sublinear scalar multiplication algorithm," in *International Conference on Cryptology in Malaysia*. Springer, 2005, pp. 171–182.

[56] V. Dimitrov, L. Imbert, and P. K. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2005, pp. 59–78.

[57] P. K. Mishra and V. Dimitrov, "Efficient quintuple formulas for elliptic curves and efficient scalar multiplication using multibase number representation," in *International Conference on Information Security*. Springer, 2007, pp. 390–406.

[58] P. Longa and A. Miri, "New composite operations and precomputation scheme for elliptic curve cryptosystems over prime fields (full version)." *IACR Cryptology ePrint Archive*, vol. 2008, p. 51, 2008.

[59] T. Chabrier and A. Tisserand, "On-the-fly multi-base recoding for ecc scalar multiplication without pre-computations," in *Computer Arithmetic (ARITH), 2013 21st IEEE Symposium on*. IEEE, 2013, pp. 219–228.

[60] R. Schroeppel, "Elliptic curve point ambiguity resolution apparatus and method," Apr. 3 2000, uS Patent 7,200,225.

[61] K. Fong, D. Hankerson, J. López, and A. Menezes, "Field inversion and point halving revisited," *IEEE Transactions on Computers*, vol. 53, no. 8, pp. 1047–1059, 2004.

[62] R. M. Avanzi, M. Ciet, and F. Sica, "Faster scalar multiplication on koblitz curves combining point halving with the frobenius endomorphism," in *International Workshop on Public Key Cryptography*. Springer, 2004, pp. 28–40.

[63] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[64] A. Miyaji, T. Ono, and H. Cohen, "Efficient elliptic curve exponentiation," *Information and Communications Security*, pp. 282–290, 1997.

[65] R. P. Gallant, R. J. Lambert, and S. A. Vanstone, "Faster point multiplication on elliptic curves with efficient endomorphisms," in *Annual International Cryptology Conference*. Springer, 2001, pp. 190–200.

[66] T. J. Park, M.-K. Lee, K. Park, and K. I. Chung, "Speeding up scalar multiplication in genus 2 hyperelliptic curves with efficient endomorphisms," *ETRI journal*, vol. 27, no. 5, pp. 617–627, 2005.

[67] Z. Liu, J. Großschädl, Z. Hu, K. Järvinen, H. Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 773–785, 2017.

[68] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2005.

[69] N. A. Mohamed, M. H. Hashim, and M. Hutter, "Improved fixed-base comb method for fast scalar multiplication," in *International Conference on Cryptology in Africa*. Springer, 2012, pp. 342–359.

[70] N. A. Saqib, F. Rodriguez-Henriquez, and A. Diaz-Perez, "A parallel architecture for fast computation of elliptic curve scalar multiplication over $gf(2^m)$," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. IEEE, 2004, p. 144.

[71] J. Petit, "Analysis of ecdsa authentication processing in vanets," in *2009 3rd International Conference on New Technologies, Mobility and Security*. IEEE, 2009, pp. 1–5.

[72] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: Ecc comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

[73] D. Hankerson, J. L. Hernandez, and A. Menezes, "Software implementation of elliptic curve cryptography over binary fields," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2000, pp. 1–24.

[74] A. Antipa, D. Brown, R. Gallant, R. Lambert, R. Struik, and S. Vanstone, "Accelerated verification of ecdsa signatures," in *International Workshop on Selected Areas in Cryptography*. Springer, 2005, pp. 307–318.

[75] N. Meloni, "New point addition formulae for ecc applications," in *International Workshop on the Arithmetic of Finite Fields*. Springer, 2007, pp. 189–201.

[76] A. Sakthivel and R. Nedunchezhian, "Improved the execution speed of ecdsa over gf (2^n) algorithm for concurrent computation." *Journal of Theoretical & Applied Information Technology*, vol. 50, no. 1, 2013.

[77] E. Guerrini, L. Imbert, and T. Winterhalter, "Randomized mixed-radix scalar multiplication," *IEEE Transactions on Computers*, 2017.

[78] T. Oliveira, J. López, D. F. Aranha, and F. Rodríguez-Henríquez, "Two is the fastest prime: lambda coordinates for binary elliptic curves," *Journal of Cryptographic Engineering*, vol. 4, no. 1, pp. 3–17, 2014.

[79] M. Brown, D. Hankerson, J. López, and A. Menezes, "Software implementation of the nist elliptic curves over prime fields," in *Cryptographers' Track at the RSA Conference*. Springer, 2001, pp. 250–265.

[80] J. López and R. Dahab, "Improved algorithms for elliptic curve arithmetic in gf (2^n)," in *International Workshop on Selected Areas in Cryptography*. Springer, 1998, pp. 201–212.

[81] T. F. Al-Somani, "High-performance generic-point parallel scalar multiplication," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 507–512, 2017.

[82] D. V. Chudnovsky and G. V. Chudnovsky, "Sequences of numbers generated by addition in formal groups and new primality and factorization tests," *Advances in Applied Mathematics*, vol. 7, no. 4, pp. 385–434, 1986.

[83] W. Pan, F. Zheng, Y. Zhao, W.-T. Zhu, and J. Jing, "An efficient elliptic curve cryptography signature server with gpu acceleration," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 111–122, 2017.

[84] H. Wang, Z. Wu, and X. Tan, "A new secure authentication scheme based threshold ecdsa for wireless sensor network." in *Security and Management*, 2006, pp. 129–133.

[85] S. B. Othman, H. Alzaid, A. Trad, and H. Youssef, "An efficient secure data aggregation scheme for wireless sensor networks," in *Information, Intelligence, Systems and Applications (IISA), 2013 Fourth International Conference on*. IEEE, 2013, pp. 1–4.

[86] V. Jariwala and D. Jinwala, "A novel approach for secure data aggregation in wireless sensor networks," *arXiv preprint arXiv:1203.4698*, 2012.

[87] A. Singh, A. K. Awasthi, and K. Singh, "A key agreement algorithm based on ecdsa for wireless sensor network," in *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*. Springer, 2016, pp. 143–149.

[88] S. Lamba and M. Sharma, "An efficient elliptic curve digital signature algorithm (ecdsa)," in *Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on*. IEEE, 2013, pp. 179–183.

[89] C. Giraud and H. Thiebeauld, "A survey on fault attacks," in *Smart Card Research and Advanced Applications VI*. Springer, 2004, pp. 159–176.

[90] 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, pp. 1–6.

[91] W. Elghazel, J. Bahi, C. Guyeux, M. Hakem, K. Medjaher, and N. Zerhouni, "Dependability of wireless sensor networks for industrial prognostics and health management," *Computers in Industry*, vol. 68, pp. 1–15, 2015.

[92] R. T. Tiburski, L. A. Amaral, E. de Matos, D. F. de Azevedo, and F. Hessel, "Evaluating the use of tls and dtls protocols in iot middleware systems applied to e-health," in *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*. IEEE, 2017, pp. 480–485.

[93] A. Nadeem and M. P. Howarth, "A survey of manet intrusion detection & prevention approaches for network layer attacks," *IEEE communications surveys & tutorials*, vol. 15, no. 4, pp. 2027–2045, 2013.

[94] A. Ometov, A. Levina, P. Borisenko, R. Mostovoy, A. Orsino, and S. Andreev, "Mobile social networking under side-channel attacks: Practical security challenges," *IEEE Access*, vol. 5, pp. 2591–2601, 2017.

[95] A. Alhothaily, C. Hu, A. Alrawais, T. Song, X. Cheng, and D. Chen, "A secure and practical authentication scheme using personal devices," *IEEE Access*, vol. 5, pp. 11 677–11 687, 2017.

[96] A. Ibrahim and G. Dalkılıç, "An advanced encryption standard powered mutual authentication protocol based on elliptic curve cryptography for rfid, proven on wisp," *Journal of Sensors*, vol. 2017, 2017.

[97] S. Bojjagani and V. Sastry, "A secure end-to-end sms-based mobile banking protocol," *International Journal of Communication Systems*, 2017.

[98] U. Rajput, F. Abbas, H. Eun, and H. Oh, "A hybrid approach for efficient privacy-preserving authentication in vanet," *IEEE Access*, vol. 5, pp. 12 014–12 030, 2017.

[99] M. Masud and M. S. Hossain, "Secure data-exchange protocol in a cloud-based collaborative health care environment," *Multimedia Tools and Applications*, pp. 1–15, 2017.

[100] A. T. Lo'ai, T. F. Somani, and H. Houssain, "Towards secure communications: Review of side channel attacks and countermeasures on ecc," in *Internet Technology and Secured Transactions (ICITST), 2016 11th International Conference for.* IEEE, 2016, pp. 87–91.

[101] C. Mann and D. Loebenberger, "Two-factor authentication for the bitcoin protocol," *International Journal of Information Security*, vol. 16, no. 2, pp. 213–226, 2017.

[102] J. P. Singh, S. Kumar *et al.*, "Authentication and encryption in cloud computing," in *Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on.* IEEE, 2015, pp. 216–219.

[103] I. F. Blake, G. Seroussi, and N. P. Smart, *Advances in elliptic curve cryptography.* Cambridge University Press, 2005, vol. 317.

[104] M. Medwed and E. Oswald, "Template attacks on ecdsa," in *International Workshop on Information Security Applications.* Springer, 2008, pp. 14–27.

[105] L. Wang, Q. Li, G. Zhang, J. Yu, Z. Zhang, L. Guo, and D. W. Zhang, "A new spa attack on ecc with regular point multiplication," in *Computational Intelligence and Security (CIS), 2015 11th International Conference on.* IEEE, 2015, pp. 322–325.

[106] P.-Y. Liardet and N. P. Smart, "Preventing spa/dpa in ecc systems using the jacobi form," in *International Workshop on Cryptographic Hardware and Embedded Systems.* Springer, 2001, pp. 391–401.

[107] Y. Dou, J. Weng, C. Ma, and F. Wei, "Secure and efficient ecc speeding up algorithms for wireless sensor networks," *Soft Computing*, vol. 21, no. 19, pp. 5665–5673, 2017.

[108] D. F. Aranha, P.-A. Fouque, B. Gérard, J.-G. Kammerer, M. Tibouchi, and J.-C. Zapalowicz, "Glv/gls decomposition, power analysis, and attacks on ecdsa signatures with single-bit nonce bias," in *International Conference on the Theory and Application of Cryptology and Information Security.* Springer, 2014, pp. 262–281.

[109] Z. Liu, D. Liu, and X. Zou, "An efficient and flexible hardware implementation of the dual-field elliptic curve cryptographic processor," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2353–2362, 2017.

[110] P.-A. Fouque, S. Guilley, C. Murdica, and D. Naccache, "Safe-errors on spa protected implementations with the atomicity technique," in *The New Codebreakers.* Springer, 2016, pp. 479–493.

[111] J. Blömer and P. Günther, "Singular curve point decompression attack," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2015 Workshop on.* IEEE, 2015, pp. 71–84.

[112] A. P. Fournaris, L. Papachristodoulou, L. Batina, and N. Sklavos, "Residue number system as a side channel and fault injection attack countermeasure in elliptic curve cryptography," in *Design and Technology of Integrated Systems in Nanoscale Era (DTIS), 2016 International Conference on.* IEEE, 2016, pp. 1–4.

[113] J. W. Bos, C. Costello, P. Longa, and M. Naehrig, "Selecting elliptic curves for cryptography: an efficiency and security analysis," *Journal of Cryptographic Engineering*, vol. 6, no. 4, pp. 259–286, 2016.

[114] I. Verbauwhede, D. Karaklajic, and J.-M. Schmidt, "The fault attack jungle-a classification model to guide you," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on.* IEEE, 2011, pp. 3–8.

[115] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.

[116] A. Barenghi, G. M. Bertoni, L. Breveglieri, G. Pelosi, S. Sanfilippo, and R. Susella, "A fault-based secret key retrieval method for ecdsa: Analysis and countermeasure," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 1, p. 8, 2016.

[117] K. Liao, X. Cui, N. Liao, T. Wang, D. Yu, and X. Cui, "High-performance noninvasive side-channel attack resistant ecc coprocessor for gf (2m)," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 727–738, 2017.

[118] D. L. K. Chuen, *Handbook of digital currency: Bitcoin, innovation, financial instruments, and big data.* Academic Press, 2015.

[119] N. Koblitz and A. J. Menezes, "The random oracle model: a twenty-year retrospective," *Designs, Codes and Cryptography*, vol. 77, no. 2-3, pp. 587–610, 2015.

[120] J. C. Ha and S. J. Moon, "Randomized signed-scalar multiplication of ecc to resist power attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems.* Springer, 2002, pp. 551–563.

[121] J. Samotyja and K. Lemke-Rust, "Practical results of ecc side channel countermeasures on an arm cortex m3 processor," in *Proceedings of the 2016 ACM Workshop on Theory of Implementation Security.* ACM, 2016, pp. 27–35.

[122] S. Bhattacharya, C. Maurice, S. Bhasin, and D. Mukhopadhyay, "Template attack on blinded scalar multiplication with asynchronous perf-ioctl calls," 2017.

[123] R. Poussier, Y. Zhou, and F.-X. Standaert, "A systematic approach to the side-channel analysis of ecc implementations with worst-case horizontal attacks," in *International Conference on Cryptographic Hardware and Embedded Systems.* Springer, 2017, pp. 534–554.

[124] M. Hedabou, P. Pinel, and L. Bénéteau, "Countermeasures for preventing comb method against sca attacks," in *International Conference on Information Security Practice and Experience.* Springer, 2005, pp. 85–96.

[125] J.-M. Schmidt and M. Medwed, "A fault attack on ecdsa," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009 Workshop on.* IEEE, 2009, pp. 93–99.

[126] Z. Liu, D. Liu, and X. Zou, "An efficient and flexible hardware implementation of the dual-field elliptic curve cryptographic processor," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2353–2362, 2017.

[127] P. Sasdrich and T. Güneysu, "Cryptography for next generation tls: Implementing the rfc 7748 elliptic curve448 cryptosystem in hardware," in *Proceedings of the 54th Annual Design Automation Conference 2017.* ACM, 2017, p. 16.

[128] S. Karati, A. Das, and D. R. Chowdhury, "Using randomizers for batch verification of ecdsa signatures." *IACR Cryptology ePrint Archive*, vol. 2012, p. 582, 2012.

[129] X. Fan, A. Otemissov, F. Sica, and A. Sidorenko, "Multiple point compression on elliptic curves," *Designs, Codes and Cryptography*, vol. 83, no. 3, pp. 565–588, 2017.

[130] V. Rožić, O. Reparaz, and I. Verbauwhede, "A 5.1 μj per point-multiplication elliptic curve cryptographic processor," *International Journal of Circuit Theory and Applications*, vol. 45, no. 2, pp. 170–187, 2017.

[131] B. Möller, "Securing elliptic curve point multiplication against side-channel attacks," in *International Conference on Information Security.* Springer, 2001, pp. 324–334.

[132] B. Möller, "Parallelizable elliptic curve point multiplication method with resistance against side-channel attacks," in *International Conference on Information Security*, 2002, pp. 402–413.

[133] D. Karaklajić, J. Fan, J.-M. Schmidt, and I. Verbauwhede, "Low-cost fault detection method for ecc using montgomery powering ladder," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011.* IEEE, 2011, pp. 1–6.

[134] T. Oliveira, J. López, and F. Rodríguez-Henríquez, "The montgomery ladder on binary elliptic curves," *Journal of Cryptographic Engineering*, pp. 1–18, 2017.

[135] X. Feng and S. Li, "A high-speed and spa-resistant implementation of ecc point multiplication over gf (p)," in *Trustcom/BigDataSE/ICESS, 2017 IEEE.* IEEE, 2017, pp. 255–260.

[136] V. Jain, R. Kumar, and Z. Saquib, "An approach towards digital signatures for e-governance in india," in *Proceedings of the 2015 2nd International Conference on Electronic Governance and Open Society: Challenges in Eurasia.* ACM, 2015, pp. 82–88.

[137] T. Zefferer, F. Golser, and T. Lenz, "Towards mobile government: verification of electronic signatures on smartphones," in *International Conference on Electronic Government and the Information Systems Perspective.* Springer, 2013, pp. 140–151.

[138] R. Khatoun and S. Zeadally, "Cybersecurity and privacy solutions in smart cities," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 51–59, 2017.

[139] B. Alhaqbani and C. Fidge, "Privacy-preserving electronic health record linkage using pseudonym identifiers," in *E-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on.* IEEE, 2008, pp. 108–117.

[140] P. Paganini, "Risks and cyber threats to the healthcare industry," INFOSEC Institute, Tech. Rep., 2014, available at http://resources.infosecinstitute.com/risks-cyber-threats-healthcare-industry/#gref.

[141] Health Care Authority, "Data breach for apple health (medicaid) managed care plan," https://www.hca.wa.gov/about-hca/apple-health-

medicaid/data-breach-apple-health-medicaid-managed-care-plan-what-clients, Washington State, 2016.

[142] J. Davis, "Ehr server hack threatens data of 14,000 ivf clinic patients," healthcare IT News, Tech. Rep., 2017, available at http://www.healthcareitnews.com/news/ehr-server-hack-threatens-data-14000-ivf-clinic-patients.

[143] B. Garrett, "Vormetric data threat report," 451 RE-SEARCH, Thales Group, Tech. Rep., 2016, available at https://www.thalesesecurity.com/.

[144] R. Gajanayake, R. Iannella, and T. Sahama, "Privacy oriented access control for electronic health records," *Electronic Journal of Health Informatics*, vol. 8, no. 2, p. 15, 2014.

[145] T. Neubauer and J. Heurix, "A methodology for the pseudonymization of medical data," *International Journal of Medical Informatics*, vol. 80, no. 3, pp. 190–204, 2011.

[146] G. Ateniese, G. Bianchi, A. T. Capossele, C. Petrioli, and D. Spenza, "Low-cost standard signatures for energy-harvesting wireless sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, p. 64, 2017.

[147] S. Ghosh, A. Majumder, J. Goswami, A. Kumar, S. P. Mohanty, and B. K. Bhattacharyya, "Swing-pay: One card meets all user payment and identity needs: A digital card module using nfc and biometric authentication for peer-to-peer payment," *IEEE Consumer Electronics Magazine*, vol. 6, no. 1, pp. 82–93, 2017.

[148] N. Kshetri and J. Voas, "Banking on availability," *IEEE COMPUTER SOCIETY*, vol. 50, no. 1, pp. 76–80, 2017.

[149] RSA, "Web threat detection trends in e-commerce," https://www.rsa.com/content/dam/pdfs/2-2017/rsa-web-threat-eCommerce-ebook.pdf, A Guide to Improve Fraud Detection and Investigation, 2017.

[150] PHISHLABS, "2017 phishing trends and intelligence report: Hacking the human," https://pages.phishlabs.com/rs/130-BFB-942/images/2017%20PhishLabs%20Phishing%20and%20Threat%20Intelligence%20Report.pdf, Ecrime Management Strategies, 2017.

[151] IPC, "E-commerce sector at risk of lethaĺddos attacks this 2017," http://ipc.ph/en/e-commerce-sector-risk-lethal-ddos-attacks-2017/, 2017.

[152] L. van der Horst, K.-K. R. Choo, and N.-A. Le-Khac, "Process memory investigation of the bitcoin clients electrum and bitcoin core," *IEEE Access*, 2017.

[153] S. S. Manvi and S. Tangade, "A survey on authentication schemes in vanets for secured communication," *Vehicular Communications*, 2017.

[154] A. Greenberg, "Hackers remotely kill a jeep on the highway," Tech. Rep., 2015, available at https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[155] R. V. Numaan Huq and M. Swimmer, "Cyberattacks against intelligent transportation systems," TrendLabs, Tech. Rep., 2017, available at https://documents.trendmicro.com/assets/white_papers/wp-cyberattacks-against-intelligent-transportation-systems.pdf.

[156] S. Manvi, M. Kakkasageri, and D. Adiga, "Message authentication in vehicular ad hoc networks: Ecdsa based approach," in *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*. IEEE, 2009, pp. 16–20.

[157] G. Sharma and S. Kalra, "A secure remote user authentication scheme for smart cities e-governance applications," *Journal of Reliable Intelligent Environments*, vol. 3, no. 3, pp. 177–188, 2017.

[158] K. Angelopoulos, V. Diamantopoulou, H. Mouratidis, M. Pavlidis, M. Salnitri, P. Giorgini, and J. F. Ruiz, "A holistic approach for privacy protection in e-government," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*. ACM, 2017, p. 17.

[159] R. Gupta, S. K. Muttoo, and S. K. Pal, "Proposed framework for information systems security for e-governance in developing nations," in *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance*. ACM, 2017, pp. 546–547.

[160] S. Joshi, "112 government websites hacked in the last 3 months," Tech. Rep., 2012, available at http://www.thehindu.com/sci-tech/technology/internet/112-government-websites-hacked-in-the-last-3-months/article2999836.ece.

[161] F. O. Doug Dingwall and T. McIlroy, "Data breach sees records of 50,000 australian workers exposed," The Canberra Crimes, Tech. Rep., 2017, available at http://www.canberratimes.com.au/national/public-service/data-breach-sees-records-of-50000-australian-workers-exposed-20171102-gzdef3.html.

[162] B. Chowdhury, M. Chowdhury, and J. Abawajy, "Securing a smart anti-counterfeit web application," *Journal of networks*, vol. 9, no. 11, pp. 2925–2933, 2014.