IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 18, 2019, accepted December 3, 2019, date of publication December 5, 2019,
date of current version December 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957943

SE-Enc: A Secure and Efficient Encoding Scheme Using Elliptic Curve Cryptography

HISHAM N. ALMAJED AND AHMAD S. ALMOGREN
Chair of Cyber Security, Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11633, Saudi Arabia

Corresponding author: Ahmad S. Almogren (ahalmogren@ksu.edu.sa)

This work was supported by the Deanship of Scientific Research, King Saud University for funding through Vice Deanship of Scientific Research Chairs.

ABSTRACT Many applications use asymmetric cryptography to secure communications between two parties. One of the main issues with asymmetric cryptography is the need for vast amounts of computation and storage. While this may be true, elliptic curve cryptography (ECC) is an approach to asymmetric cryptography used widely in low computation devices due to its effectiveness in generating small keys with a strong encryption mechanism. The ECC decreases power consumption and increases device performance, thereby making it suitable for a wide range of devices, ranging from sensors to the Internet of things (IoT) devices. It is necessary for the ECC to have a strong implementation to ensure secure communications, especially when encoding a message to an elliptic curve. It is equally important for the ECC to secure the mapping of the message to the curve used in the encryption. This work objective is to propose a trusted and proofed scheme that offers authenticated encryption (AE) for both encoding and mapping a message to the curve. In addition, this paper provides analytical results related to the security requirements of the proposed scheme against several encryption techniques. Additionally, a comparison is undertaken between the SE-Enc and other state-of-the-art encryption schemes to evaluate the performance of each scheme.

INDEX TERMS Asymmetric cryptography, authenticated encryption, elliptic curve cryptography, encryption, signature.

I. INTRODUCTION

The growing need to maintain data confidentiality and integrity has resulted in an explosion of interest in cryptography schemes [1], [2]. Schemes used to encrypt data are divided into two types: namely, symmetric and asymmetric cryptography. The first type, which uses a single secret keyword to encrypt and decrypt the data, is useful when the sender and recipient agree on a shared secret key before sending the data. However, if both parties cannot find a secure way to exchange this secret key, then the need for asymmetric cryptography arises [3].

A. ASYMMETRIC CRYPTOGRAPHY AND ENCRYPTION ATTACKS

In contrast to symmetric cryptography, asymmetric cryptography, also known as public-key cryptography, uses pairs of keys: namely, a public key and a private key [4]. The sender uses the public key to encrypt the data, thereby meaning that this key should be available for use by any party. In the case of the private key, it is used to decrypt the encrypted data, and thus it is mandatory that it is known only to the recipient. This protocol solves the need to exchange keys between two parties in a secure way. However, there is a drawback associated with the use of asymmetric cryptography, which is the size of keys and the computation needed to encrypt and decrypt a message. Therefore, Elliptic Curve Cryptography (ECC) become widely used to solve the issues of key sizes and computation overhead where low end devices need to maintain performance [5], [6]. The ECC provides the same level of security as the Rivest-Shamir-Adleman (RSA) algorithm with short keys [7], [8].

Cryptography is vulnerable to many well-known attacks that threaten the encryption process of these schemes. Examples include the known-plaintext attack (KPA), chosen-plaintext attack (CPA), ciphertext-only attack (COA), and chosen-ciphertext attack (CCA). The first attack, KPA, can occur when an adversary has the ability to obtain the plaintext

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

and its corresponding ciphertext, as a consequence of which the adversary attempts to obtain the secret key [9]. The second attack, CPA, can occur when an adversary has the ability to choose random plaintexts transmitted for encryption, and then to obtain the corresponding ciphertexts. Therefore, the adversary in the CPA attempts to reduce the security of the scheme [10]. In the third attack, COA, the adversary is assumed to have access only to a set of ciphertexts, meaning that they can extract the plaintext or the secret key [11]. Finally, the CCA is characterised by an adversary's attempt to acquire information from plaintexts by obtaining the decryptions of selected ciphertexts. Therefore, the adversary attempts to obtain the secret key used to decrypt the message [12]. Throughout this work, the tests used against each of these attacks are denoted as follows [13]:

- IND-CPA: Indistinguishable under chosen plaintext attack
- IND-CCA: Indistinguishable under chosen ciphertext attack
- IND-CCA1: Indistinguishable under non-adaptive chosen ciphertext attack
- IND-CCA2: Indistinguishable under adaptive chosen ciphertext attack
- IND-CCA3 [14]: Indistinguishable authenticated adaptive chosen ciphertext attack

B. OUR CONTRIBUTION

The motivation behind this work is the security flaws that many schemes illustrated in previous section may vulnerable to them. Many papers lack of the description how the encoding phase is actually done. Many security analysis need to be addressed in order to provide an Authenticated Encryption scheme namely KPA, CPA, CCA, etc. The main focus and contribution of this work is to offer an AE scheme using ECC as following:

- Describe the security flaws in ECC encoding and mapping phases. These flaws fall under several encryption attacks such as COA and CPA.
- Secure message encoding phase by applying Block Cipher Modes of Operation that resistant to COA, KPA, CPA, Replay Attack and Malleability Attack.
- Provide a comprehensive study of the padding step that significantly affect the performance of the scheme in the encoding phase.
- Provide security analysis for the proposed scheme that satisfying the security requirements in the second item.

Having provided an introductory overview, the remainder of this paper is organised as follows: in Section 2, a literature review of other schemes is provided. Following by the preliminaries section which covers the problem statement and overview of ECC. Section 4 covers in details relating to the SE-Enc scheme are presented; in Section 5, security analysis and performance evaluations are described in detail; and finally, Section 6 offers concluding remarks and discusses avenues for future research.

II. RELATED WORKS

Several schemes advocate the use of ECC to exchange and secure communications between two parties. For instance, the elliptic curve integrated encryption scheme (ECIES) uses ECC to generate a shared key between two parties, and then encrypts the message using an AE scheme [15]–[18]. However, several schemes have been proposed to reduce the computation overhead needed to calculate the public and private keys. Other have schemes introduced methods to encode the message numerically, for instance using ASCII code [19], converting it into bits, and then into decimals or by creating a private mapping table. Significantly, these schemes are vulnerable to several attacks, including various types of plaintext attack. Therefore, other schemes have been formulated to overcome these vulnerabilities, with notable methods relying on the XOR operation and mapping to a secret matrix. However, these schemes are also vulnerable to ciphertext attacks, collision attacks, and the MITM attack. Many schemes do not offer AE where it should be invulnerable and signed.

Recent proposed schemes are using ECC to reduce the encryption computation to overcome the limitation of low-end devices. Nearly all of these schemes have not provided details about how messages are converted into numerical values to be mapped to the elliptic curve. These schemes provide many enhancements in certain areas other than the security enhancements on the ECC itself. For instance, [20]–[22] proposed schemes that use ECC without detailing how messages are encoded and mapped to the selected curve. As a result, these schemes reduce the computation process and power consumption. On the other hand, [23]–[25] provided efficient algorithms that enhance scalar multiplication on the elliptic curve. However, several schemes have given details about the encoding and mapping phases. The remainder of this literature review focuses on these schemes and, in particular, the question of how the message is converted into numerical values to be mapped on the elliptic curve.

Koblitz [26] introduced the first curve used in ECC. In addition, the author defined how to encode plaintext to numerical values to be mapped to his curve. In his scheme, the author used each character in the plaintext and encoded it based on the ASCII table. For instance, for the word "Hello", the outcome after encoding to numerical values was "72 101 108 108 111". Following this, the author mapped each ASCII value to the elliptic curve. Given that the ASCII table is common and known to all parties, there is no need to distribute it to another party for the purpose of decrypting and decoding the message. However, an adversary can learn from the ciphertext transmitted between the two parties because the same encrypted characters are repeated in the ciphertext. Therefore, an adversary could launch a ciphertext attack and decrypt the transmitted message.

Tiwari and Kim [27] proposed a novel method informed by DNA-based ECC. DNA genome sequences were used to assign values to different character sets in a message. Since this mapping is random and uses pseudo-random data, the randomisation of mapping characters is employed to encrypt the message and, furthermore, to ensure that it cannot be decrypted without the DNA genome sequences. However, both the sender and the receiver must agree on these sequences before encrypting and decrypting the message. Therefore, the DNA genome sequences need to be secret and no parties except the sender and the recipient can use these. Otherwise, the scheme becomes vulnerable to encryption flaws. In addition, the scheme proposed by these authors fails to address the encoding process that converts the message into numerical values to be used for mapping to the DNA genome sequence. Consequently, the scheme is vulnerable to certain encryption attacks.

Singh and Singh [28] proposed an image encryption scheme using ECC and reduced the encryption computation to the following two operations: firstly, an operation involving pixel grouping into a single integer, where the number of pixels in each group is based on the ECC key size; and secondly, an operation in which these groups of pixels are mapped to one large integer, which is then mapped to the elliptic curve. The performance of this scheme depends on the number of pixels that exist in one group. Significantly, a large number of pixels in one group decreases the computation overhead and, in this way, increases the performance. However, the number of pixels in one group depends on the key size, where large keys increase the pixel count. While this may be true, incrementing the key size leads to an increase in the encryption and decryption computation and storage overhead. As a result, the performance is affected in both cases.

Sengupta and Ray [29] introduced message mapping and reverse mapping on ECC. In their scheme, the authors first selected a fixed number of characters from the message for every step. Following this, the second step involved mapping this set of characters to the ASCII code. In turn, the third step mapped this ASCII code to the elliptic curve. In order to prevent non-mapping results, the authors proposed padding each set of characters to 8 bits to add one bit each time the mapping failed to find corresponding y value. Although the scheme produced promising results, analysis indicates that is vulnerable to CPA when there is an equality of the set of characters. In addition, the scheme did not specify the encryption step required to secure the ciphertext.

Singh and Singh [30] and Das and Giri [31] proposed a scheme to encrypt text messages using ECC. The scheme starts by converting the message into its corresponding ASCII value using the ASCII table. Then, the result is partitioned to a fixed number of characters, which results in a set of groups. Following this, each group is converted into large integer values, taking the base as 65536. If the previous step count is odd, then padding is used with the ASCII blank space code (i.e., “32”) to ensure that it is even, which results in P_m. In turn, a random number k is selected and kG is computed, where G is the base point, along with kP_b, where P_b is recipient’s public key. Following this, the ciphertext $(kG, P_m + kP_b)$ is computed. This scheme uses the ASCII table to encode messages, which makes it vulnerable to CPA. In addition, the authors did not discuss how P_m is mapped to the elliptic curve.

Similarly, Das and Giri [31] proposed encoding two algorithms to create group of numerical values using sum of positional weight with base b $IntegerDigits[n, b]-1$. The first algorithm of encoding when the b is dynamic integer number. The highest value of this number is the highest value of ASCII table which is 65536. In addition, the value n is key size used on the scheme where they suggest to use 192 bit key. Thus, the number of the groups that can be combined according to there method is $IntegerDigits[192bit, 65536] - 1 = 11$. Authors suggest that the b could be reduce to less than 65536, thus the number of groups can be increased to more than 11. However, authors did not provide any details about how to reduce the ASCII table to reduce the base. The second algorithm when the b is not dynamic, in the other word it is fixed. Authors suggest that the number of combing group is equal to $\frac{Number\ of\ p\ digits}{IntegerDigits[n,b]-1}$ which equal to $\frac{58-1}{11} = 6$. Both algorithms provide small combing groups which result on increase of computation overhead. In addition, both algorithms use ASCII values without manipulating it, which result to both algorithms are vulnerable to CPA.

King’s [32] scheme mapped a message to an elliptic curve using a probabilistic method. In this scheme, the author used the binary string interpretation of a message for the probabilistic equation. Thus, for each $j, j = 0$ to $K - 1$, the x_j of the elliptic curve point is equal to $M * K + j \ mod \ p$, where M satisfies $(M + 1)K < p$. In turn, y_j computed as $x_j^3 + ax_j + b$, and if y_j has a square root, then the process stops. If y_j has no square root, then the scheme continues computing to the next x_j until one is found. For some cases, the j becomes larger than $K-1$, resulting in a situation where M cannot be mapped to the elliptic curve. The principal drawback of this scheme is the question of how to agree on the value of K such that the recipient can decrypt the message. In addition, when M is repeated several times, there is a chance that an adversary can learn the value of the repeated cipher, which is known as the chosen ciphertext attack.

III. PRELIMINARIES

A. THE PROBLEM STATEMENT

Due to its effectiveness, we believe that ECC is the future for securing communications in low computation devices such as IoT devices and WSNs. Several schemes have been proposed in the literature to secure nodes with low-capability resources using ECC. However, these schemes suffer from serious security flaws owing to weaknesses in the ECC implementation. These limitations include using the wrong EC, relying on weak message encoding, and facilitating weak message mapping to the EC. Moreover, many schemes fail to provide integrity to the encrypted message, thereby meaning that they do not offer effective AE.

With these considerations in mind, it is necessary to formulate a scheme that offers confidentiality and integrity for the communications that occur between parties on a network. Furthermore, the scheme should be proofed to ensure robust levels of security against several encryption attacks, including plaintext and ciphertext attacks. It is equally critical that the scheme should be suitable for low computation and limited resource devices, and it should perform adequately with respect to the minimization of processing time and storage capacity.

In this research proposal, the following questions have been established to define the problem:

- Does the proposed scheme offer confidentiality and integrity with respect to AE?
- Can the proposed scheme resist encryption attacks and provide negligible propriety to the challenging game with the encryption and decryption oracle, for instance, IND-CPA, IND-CCA, IND-CCA1, etc?
- Does the proposed scheme outperform similar schemes?

B. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

ECC is used widely in low computation devices such as wireless sensor networks (WSNs) and Internet of things (IoT) devices. This is due to its superiority in providing the same level of cryptographic hardness as other public key protocols with very small size keys and a low computation overhead. For instance, encryption using the RSA algorithm with a 1024-bit key is equal to ECC encryption with a 160-bit key. The substantial difference in key sizes means that devices with lower computational capabilities will perform more effectively [33]. Significantly, ECC is based on the discrete logarithm structure of elliptic curves over finite fields. ECC is used to exchange keys and to facilitate secure communications between two parties, and it offers a way to sign messages to maintain integrity and prevent forgery. Many schemes use ECC to secure communications, but these schemes differ from several areas of encryption. As previously noted, certain schemes are used only to facilitate key generation, while others are used both to encrypt and sign messages.

Elliptic curve over $\mathbb{Z}_p,\ p > 3$ is the set of all pairs $(x, y) \in \mathbb{Z}_p$ such that:

$$y^2 \equiv x^3 + a\ .\ x + b \ mod\ p \tag{1}$$

where $a, b \in \mathbb{Z}_p$ and $4.a^3 + 27.b^2 \neq 0\ mod\ p$. Therefore, an elliptic curve should be nonsingular, which means that the plot has no self-intersections or vertices. Figure 1 shows an example of the elliptic curve $y^2 = x^3 - 3x + 3$.

Group operations on an elliptic curve are denoted as addition '+'. For instance, let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. Then the $P + Q = (x_1, y_1) + (x_2, y_2) = (x_3, y_3)$. For a special operation when $P = Q$, then $P+P = (x_1, y_1)+(x_1, y_1) = 2P)$ is referred to as point doubling. Figure 2 depicts an example of elliptic curve point addition. Figure 3 depicts an example of elliptic curve point doubling.

Group multiplication is the ECC major operation [34]–[37]. It is defined as the number of instances of group point doubling. The group multiplication consists of d, an integer number known as a private key, and $G = (x_i, y_i)$, a base point on the elliptic curve. Therefore, the operation

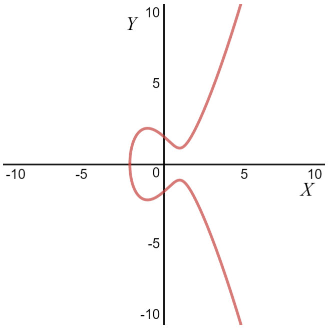

FIGURE 1. A genuine elliptic curve $y^2 = x^3 - 3x + 3$ over $\mathbb{Z}_p$.

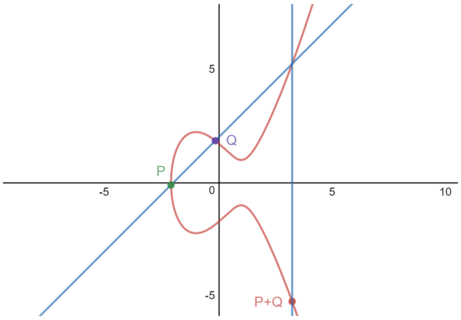

FIGURE 2. Elliptic curve point addition for $P + Q$.

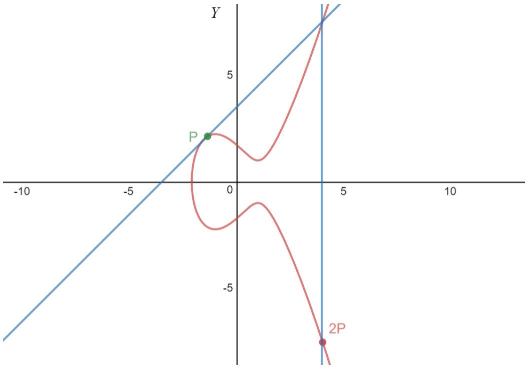

FIGURE 3. Elliptic curve point doubling for $P + P = 2P$.

dG is the d doubling time for G, which results in another point (x_j, y_j), known as the public key. The security of ECC is based on the hardness of the mathematical problem [38], which indicates that in knowing the public key point and the base point G, it is impossible to find d in polynomial time. This is referred to in the literature as the elliptic curve discrete logarithm problem (ECDLP) [39].

ECC consists of several phases to exchange keys and secure communications [40]–[42]. These phases are used separately and/or together, and they can be listed as follows:

- Generating parameters such as defining the elliptic curve, and calculating P_r and P_u
- Numerical encoding of the message (for encryption)
- Hashing the message (for signing)
- Mapping the encoded message to an elliptic curve

The main computation involved in the first phase is the calculation of the public key. In ECC, the public key can be derived as the product of d (i.e., the private key) and G (i.e., the base point) [43]–[46]. This multiplication can be improved in several ways by enhancing scalar multiplication in the elliptic curve. The second phase is concerned with the technique used to convert the message characters into numbers, since ECC encryption deals with numbers [47]. Thus it is important to encode each message in a way that prevents any encryption attacks such as the abovementioned plaintext attacks or ciphertext attacks. Similarly, the third phase involves signing the message to ensure that the sender is the one who sent the message, and to safeguard against external modification. The final phase is concerned with mapping the encoded and signed message to the elliptic curve [48]. It is important to map the message to the curve in a way that prevents encryption attacks, and thus it is necessary to combine the second and fourth phases in any proposed scheme to ensure that the encryption properties hold. In addition, signing the encrypted message is a major step that many schemes overlook. When signing an encrypted message, this completes the integrity and confidentiality of the message, thereby qualifying it as an authenticated encryption (AE) scheme.

1) ENCODING THE MESSAGE TO NUMERICAL VALUES

Various schemes have proposed approaches by which messages can be encoded to numerical values, thereby allowing them to be mapped to an elliptic curve. These schemes use the ASCII table to convert each character into its corresponding decimal number [31], [40], [49]. For example, the letter ‘a’ is encoded to 97, ‘b’ to 98, and so on. The principal flaw with this method is that it falls under the plaintext attack, since the ASCII table is known to everyone. Therefore, another scheme was introduced which relies on a matrix-based approach [50]. This scheme maps a character to decimal numbers which are unknown to all parties except for the recipient. However, two issues are associated with the matrix-based approach: firstly, the question of how to deliver the table to the recipient in a secure way, thereby preventing plaintext attack; and secondly, it is notable that if the table is delivered securely, then the encrypted message will fall under the ciphertext attack, since the encrypted characters are repeated as plain characters.

A third scheme has been proposed, in which the first character ASCII code undergoes XORing with the initial vector IV [51]. XORing is then applied to the second character with previous results, and the process is iterated over. Significantly, this scheme falls under the plaintext attack once the IV is known. As for the fourth scheme, this proposes the multiplication of the ASCII code with a number, where the number is agreed on by the sender and receiver [52]. However, similar to the second scheme, the issue of establishing agreement between the two parties is complex, and this scheme could also fall under the ciphertext attack.

Each of these schemes is vulnerable to a tampering attack, where an adversary can modify the ciphertext without being detected by the recipient. The encrypted message only provides confidentiality and does not support integrity by itself. Therefore in order to maintain the integrity of the encrypted message and ensure that the ciphertext is encrypted by the AE scheme, it needs to be signed before transmission.

2) MAPPING THE MESSAGE TO AN ELLIPTIC CURVE

Once the message has been encoded, it then needs to be mapped to the agreed-upon elliptic curve. Mapping occurs when the encoded message is assigned to the parameter x_i and there exists a corresponding y_i, such that $(x_i, y_i) \in E_p(a, b)$. If there is no corresponding y_i, then x_i is incremented by 1 until a corresponding y_i is identified [53]–[55]. Thus, many schemes append certain bits to the message to avoid changing x_i, attempting to find the corresponding y_i. Nevertheless, it is worth noting that even finding a corresponding y_i is an overhead to every scheme. Hence, many schemes simply ignore the y_i as it does not play a role in decryption.

When a scheme provides effective encoding that assures the required encryption properties, there is a chance that security flaws occur when engaging in improper mapping. Therefore, an effective scheme should provide viable and complementary encoding and mapping mechanisms. In fact, encryption in ECC amounts to encoding and mapping, and so these elements should always be together and never be separated.

3) SIGNING THE ENCRYPTED MESSAGE FOR AUTHENTICATED ENCRYPTION

Information security consists of three issues: namely, confidentiality, integrity, and availability (CIA) [56]–[58]. Encryption assures the first part, but by itself, it cannot guarantee integrity. Therefore, in order to maintain integrity, it is necessary to assign a value to the message or ciphertext. This value can be obtained by computing the message using one-way hash functions, and sending it to the recipient to assure the integrity of the received message. Using this process, we can add the sender’s private key to the one-way hash function to maintain the integrity of the message and ensure non-repudiation [59]–[61]. A recipient can verify that the signer actually sent the message by undertaking the same computation with the signer’s public key instead of the private key. A signature by itself cannot guarantee confidentiality, but it ensures that received message is not tampered with by an unauthorised user.

As a well-known method used to sign a message for low computation devices, ECDSA is effective in providing small keys (e.g., 160-bit keys) for signatures with the same level of cryptographic hardness as the RSA algorithm with 1024 bits. In order to sign a message m using EC, it is necessary for the signer to undertake the following steps [62]–[64]:

- Compute r, where r is the $X_R \ mod \ p$ of $R = k * G$, and where k is random number and G is base point
- Compute s, where $s \equiv (h(m) + d * r)k^{-1} \ mod \ p$, and where d is the signer’s private key
- Send $(m, (r, s))$

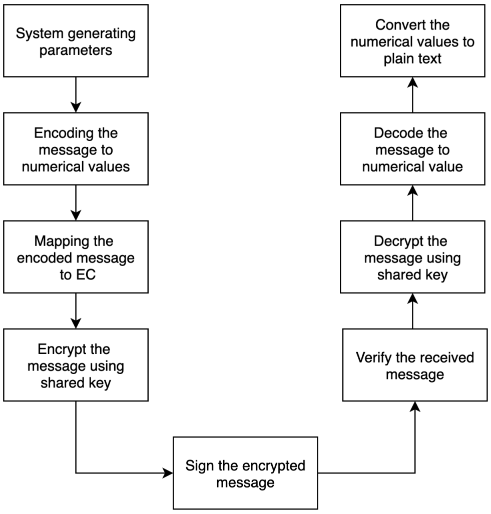
System generating parameters
Encoding the message to numerical values
Mapping the encoded message to EC
Encrypt the message using shared key
Sign the encrypted message
Verify the received message
Decrypt the message using shared key
Decode the message to numerical value
Convert the numerical values to plain text

FIGURE 4. High-level view of the proposed scheme.

IV. THE PROPOSED SCHEME (SE-ENC)

The SE-Enc scheme consists of nine phases: generating system parameters; encoding the message, mapping to the elliptic curve; encrypting the mapped points; signing the encrypted message; verifying the received message; decrypting the message; decoding the decrypted message; and converting the decoded message into plaintext. The main focus and contribution of this work is to offer an AE scheme using ECC with secure message encoding, mapping, and encryption. In addition to these phases, it is noteworthy that the first phase is a major step, and many proposed schemes neglect to consider the importance of having a shared key between the two parties to encrypt the message. This phase was included in the SE-Enc scheme because it can reasonably be viewed as a major phase for any system that needs to facilitate AE. In Figure 4, the nine phases of the proposed scheme are illustrated.

A. GENERATING SYSTEM PARAMETERS

The main advantage of this phase is the generation of the shared secret key between the two parties. This key is used to encrypt the mapped points on the elliptic curve. Table 1 illustrates the system generation notations used in the SE-Enc scheme for each session.

It is necessary for the sender to create a shared session key k_{sh} in order to encrypt mapped points on the elliptic curve. Thus, using their private key d_s, as well as the recipient's public key PU_r, the sender can generate k_{sh}. The process of generating this key is illustrated in Figure 5.

Benefiting from ECDLP, both the sender and the recipient can agree on a shared key. We can describe this process in the following algorithm:

B. ENCODING THE MESSAGE TO NUMERICAL VALUES

In this work, the encoding and mapping mechanism steps from [29] are extended to overcome the security flaws associated with the current scheme. Each message is divided into several blocks B, where each block B contains N characters. The following equation is used to calculate N:

$$N \leq \left\lfloor \frac{p-8}{8} \right\rfloor \tag{2}$$

Thus, in the SE-Enc scheme, the value of N is equal to 23, where $p = 192$. Similarly, the number of blocks B required is obtained by dividing the total number of characters in M by N. for instance, for a message with 1000 characters, the number of blocks is equal to 44. This can be achieved by using the following equation:

$$B = \left\lceil \frac{M}{N} \right\rceil \tag{3}$$

The rationale for dividing the message to this length stems from the need to encrypt each mapped point to the same length as p. In addition, one character is removed from each block to pad it with the 3 bits of zeros that are necessary for the mapping phase. Figure 6 depicts the steps required to encode the message to numerical values. For each message, after obtaining the blocks for the message M, each set of charters in each block is converted to its binary value. Afterward, the first block of binary values are XORed with the IV. Accordingly, each following blocks are XORed with previous XORed block. Finally, each XORed block is padded with 3bits for mapping in the next phase.

Algorithm 2 describes the steps involved in encoding the messages before mapping to the elliptic curve.

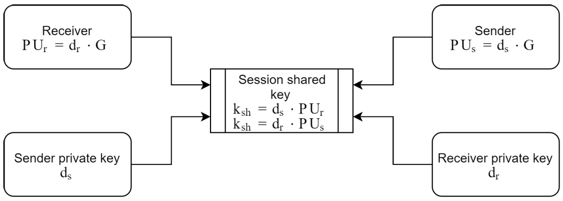

FIGURE 5. Creating a shared key for the sender and the recipient.

TABLE 1. List of notations used to generate scheme parameters.

Notation	Description of notations
d_s	Sender private key
d_r	Recipient private key
G	Base point on elliptic curve
PU_s	Sender public key = $d_s * G$
PU_r	Recipient public key = $d_r * G$
p	Large prime number (192-bit)
a, b	EC coefficients, s.t. $4a^3 + 27b^2 \; mod \; p \neq 0$
$y^2 \equiv x^3 + ax + b \; mod \; p$	EC equation to map points to EC
H	Hash function to sign the message C_M
k_{sh}	Shared session key
M	Total number of characters in the message
B	Number of blocks for each message
N	Number of characters on each block
IV	Random initial vector (192-bit)
k	Randomly securely selected from $[1, p-1]$
C_M	The encrypted message (all encrypted points)

Algorithm 1 Key Agreement Algorithm Between Sender and Recipient

```
Input: $PU_r,\ d_s,\ PU_s,\ d_r$
Output: Shared key $k_{sh}$
Sender : apply following multiplication $d_s * PU_r$;
Recipient : apply following multiplication $d_r * PU_s$;
Both multiplications are equal;
$k_{sh} \leftarrow$ the results;
```

Algorithm 2 Message-Encoding Algorithm

```
Input: The message $M$ and $p$
Output: Encoded message
Sender : obtain the $M$ message;
Sender : obtain the $N$ number of blocks;
Sender : obtain the $B$ number of blocks;
Sender : divide the message into $B$ blocks;
Sender : divide the block into $N$ characters;
Sender : convert each char to binary;
Sender : XOR first block with $IV$;
Sender :
for each block XOR it with previous XORed block;
Sender : pad 3bits of zeros into each XORed blocks;
Encoded message $\leftarrow$ the results;
```

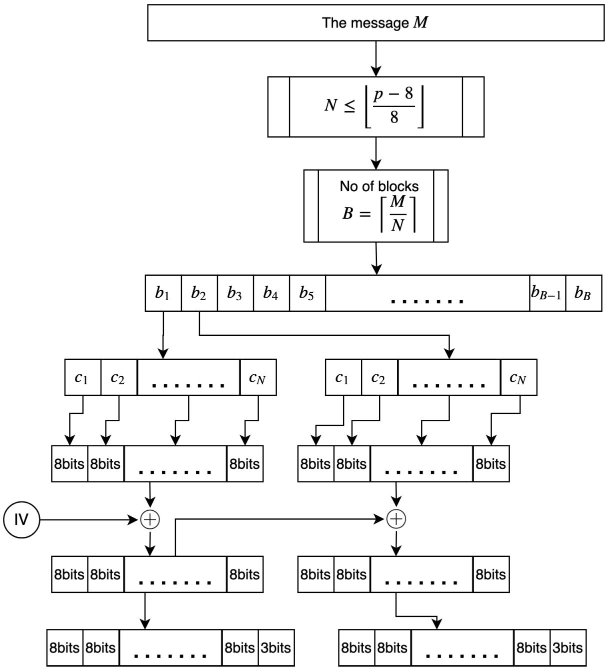

FIGURE 6. Encoding the message.

C. MAPPING THE MESSAGE TO AN ELLIPTIC CURVE

Mapping a message to an elliptic curve means that (x_i, y_i) satisfies the elliptic curve given in equation 1. Therefore, it is necessary to find the y_i value which corresponds to x_i for each point. For each block from the encoded message, the block containing a binary value is converted into a decimal value. In our proposed scheme, each block from the encoded message, the block containing a binary value is converted into a decimal value. Following that, using the generated EC equation we map this value to EC by find the correspond y_i. Figure 7 depicts the process of mapping an encoded message to an elliptic curve.

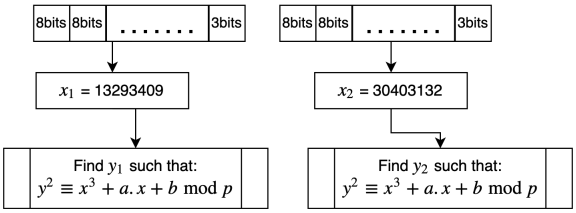

FIGURE 7. Mapping encoded message to elliptic curve.

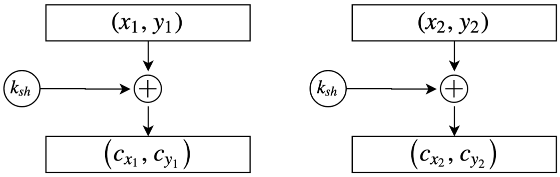

FIGURE 8. Encrypting the mapped points using shared key.

Algorithm 3, presented below, provides an account of the steps needed to map the message.

Algorithm 3 Mapping Encoded Block to EC Algorithm

```
Input: Encoded block
Output: Mapped points
1 Sender :
  obtain the decimal value for the encoded block;
2 Sender : obtain $y_1$ from the EC equation;
3 Sender : if $x_1$ cannot have corresponding $y_1$;
4 Sender : increment $x_1$ by 1;
5 Sender : repeat steps $2 - 4$ until find $y_1$;
6 Mapped points $\leftarrow$ the results;
```

D. ENCRYPTING THE MAPPED POINTS

Many schemes overlook the encryption phase and assume that the mapping phase is sufficient to secure the message. However, this view is not correct, since mapping points to an elliptic curve means that the points are eligible to be multiplied with the private key to gain the ECDLP hardness. There are several ways to secure these points. In the SE-Enc scheme, these points are encrypted by adding each point to k_{sh}. Consequently, it is cryptographically hard to retrieve the mapped points without the shared key. Figure 8 illustrates the steps involved in the process of encrypting the mapped points.

The algorithm used to encrypt the mapped points using the shared key k_{sh} is given as follows:

Algorithm 4 Encrypting Mapped Points Using Shared Key Algorithm

```
Input: Mapped points, $k_{sh}$
Output: Encrypted points
1 Sender : obtain $k_{sh}$;
2 Sender : add $k_{sh}$ to the mapped point;
3 Sender : repeat step 2 for all mapped points;
4 Encrypted points ← the results;
```

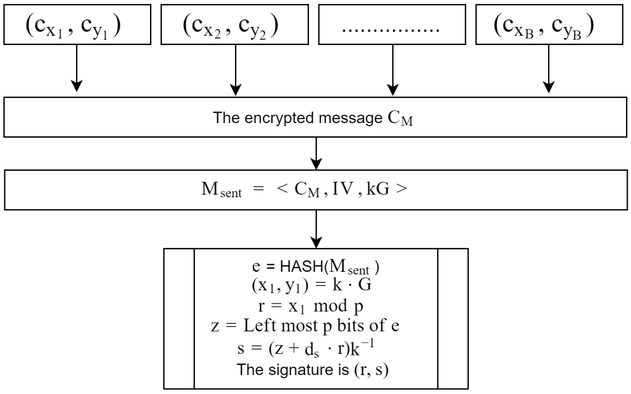

FIGURE 9. Signing the encrypted points by sender d_s.

E. SIGNING AND VERIFYING THE ENCRYPTED MESSAGE

AE schemes assure confidentiality and integrity of the transmitted message between two parties. In the SE-Enc scheme, confidentiality is maintained throughout the previously mentioned phases. To maintain integrity, the sender signs the encrypted points using ECDSA.

1) SIGNING THE ENCRYPTED MESSAGE

Each encrypted message C_M consists of all encrypted points that are mapped to the elliptic curve. In addition, IV, kG are included with C_M to the sent message M_{sent}. Thus, to produce an authenticated encryption message, the proposed scheme applies ECDSA to M_{sent}. The signing process depicted in Figure 9.

The following algorithm describes how to sign the C_M:

Algorithm 5 Signing the Encrypted Points Algorithm

```
Input: The Sent Message $M_{sent}$
Output: The signed encrypted points $(r, s)$
1 Sender : obtain $e = HASH(M_{sent})$;
2 Sender : obtain $z = leftmostp$ bits of $e$;
3 Sender : select $k$;
4 Sender : obtain $r = x \bmod p$ where $x$ is $(x, y) = k.G$ and $r \neq 0$;
5 Sender : if $r == 0$ go step 3;
6 Sender : obtain $s = (z + d_s * r)\, k^{-1}$ if $s == 0$ go step 3;
7 Sender : the pair $(r, s)$ is the signature;
8 $(r, s)$ ← the results;
```

2) VERIFYING THE SIGNED MESSAGE

The recipient can verify and decrypt the signed received message $< M_{sent}, (r, s) >$ using the sender's public key. Figure 10 describes the process of verification engaged in by the recipient.

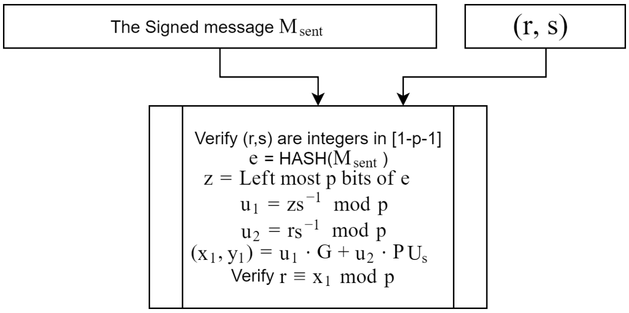
The Signed message M_{sent}
(r, s)
Verify (r,s) are integers in [1-p-1]
e = HASH(M_{sent})
z = Left most p bits of e
$u_1 = zs^{-1} \bmod p$
$u_2 = rs^{-1} \bmod p$
$(x_1, y_1) = u_1 \cdot G + u_2 \cdot PU_s$
Verify $r \equiv x_1 \bmod p$

FIGURE 10. Verify the message by the recipient.

Algorithm 6 describes how the recipient verifies the sender and the integrity of the message.

F. DECRYPTING THE MESSAGE

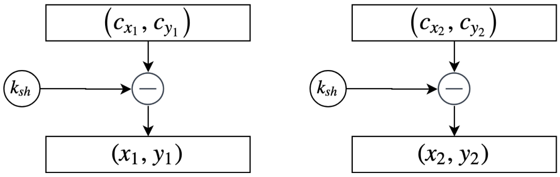

FIGURE 11. Decrypting mapped points using shared key.

The following phases are the reverse of the previous phases. This is because the decryption phase is by definition a reversal of the encryption phase. In the SE-Enc scheme, the property of reversibility for the encryption phase is verified. This is because in ECC, subtraction is the inverse operation of addition. Therefore, to decrypt the encrypted points, the recipient must subtract it with k_{sh}. Figure 5 previously illustrated how the recipient generates k_{sh}, and Figure 11 indicates how to decrypt the encrypted points.

Similar to the encryption phase, the mapped points are obtained from the encrypted points using the shared key k_{sh}. This phase is described as follows:

G. DECODING THE DECRYPTED MESSAGE

Following the completion of the decryption phase, the output is a set of mapped points. These points consist of two pairs, represented by x_i and y_i. The y_i value is only used in mapping the points to the elliptic curve. Therefore, the decoding phase is simply concerned with x_i, which is used to represent the binary values that are employed in the converting to plaintext phase. The steps involved in the decoding phase are illustrated in Figure 12.

The steps used to decode the mapped points are described in the following algorithm:

Algorithm 6 Verifying the Message by the Recipient Algorithm

```
Input: $M_{sent}$, $(r, s)$
Output: Verified message
1 Recipient : verify $(r, s)$ are integers in $[1, p-1)]$;
2 Recipient : obtain $e = HASH(M_{sent})$;
3 Recipient : obtain $z =$ leftmostp bits of $e$;
4 Recipient : obtain $u_1 = es^{-1}$ mod $p$;
5 Recipient : obtain $u_2 = rs^{-1}$ mod $p$;
6 Recipient : calculate $(x_1, y_1) = u_1 * G + u_2 * PU_s$;
7 Recipient : verify $r \equiv x_1$ mod $p$;
8 Verified message ← the results;
```

Algorithm 7 Decrypting Encrypted Points Using Shared Key Algorithm

```
Input: Encrypted points, $k_{sh}$
Output: Mapped points
1 Recipient : obtain $k_{sh}$;
2 Recipient : subtract $k_{sh}$ from the encrypted point;
3 Recipient : repeat step 2 for all encrypted points;
4 Mapped points ← the results;
```

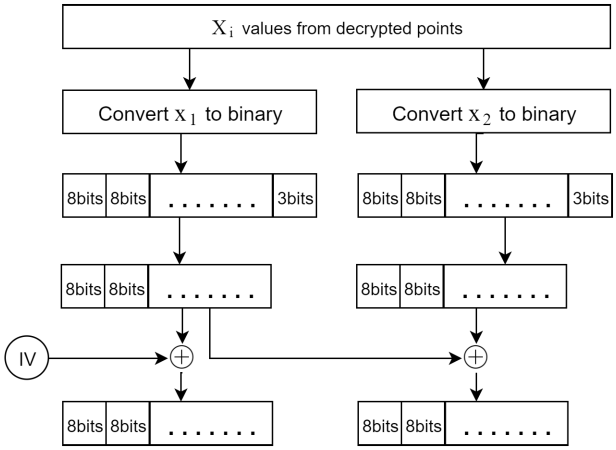

FIGURE 12. Decoding x_i to binary values.

H. CONVERTING THE DECODED MESSAGE TO PLAINTEXT

The final phase is concerned with converting the binary values into their corresponding characters. These characters represent the plaintext message M. Figure 13 shows the steps needed to convert the binary values into a plaintext message.

Algorithm 9, given as follows, describes how to convert the binary values to plaintext:

V. SECURITY ANALYSIS AND PERFORMANCE EVALUATIONS

The main objective of this work was to provide an AE scheme. Therefore, it is important to conduct a security analysis of the proposed SE-Enc scheme. In fact, from the perspective of many researchers, security analysis is the critical aspect of any cryptography scheme. However, in order to provide a feasible scheme, each scheme should be applicable to perform reasonable computation. Accordingly, we comparatively examine the performance results of the SE-Enc scheme in relation to other ECC schemes.

As noted previously, the encryption tests used to evaluate cryptography schemes are IND-CPA, IND-CCA, IND-CCA1, IND-CCA2, and IND-CCA3. It should be noted that any scheme evaluated against IND-CCA3 implies that it is also verified against IND-CCA2, IND-CCA1, and IND-CPA. Thus we evaluated SE-Enc against IND-CCA3.

Algorithm 8 Decoding Mapped Points to Binary Values Algorithm

```
Input: Mapped points
Output: Encoded block
1 Recipient : obtain $x_i$ value for the mapped point;
2 Recipient : convert $x_i$ to the binary value;
3 Recipient : remove the padding 3bits for each block;
4 Recipient : XOR first block with IV;
5 Recipient :
  for each block, XOR it with previous block;
6 Recipient : repeat step 4 for all blocks;
7 Binary values ← the results;
```

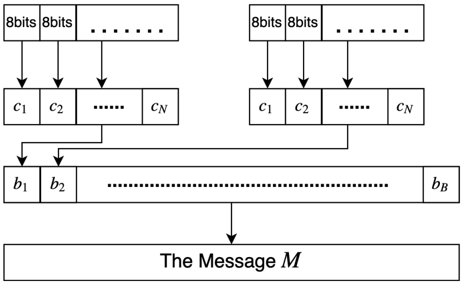

FIGURE 13. Converting binary values into plain text message M.

A. AUTHENTICATED ADAPTIVE CHOSEN CIPHER TEXT ATTACK (IND-CCA3)

A scheme is indistinguishable under IND-CCA3 when an adversary gives the power to submit any $m_{i,0}$, $m_{i,1}$ $i = [1, 2, \ldots, q]$ encryption queries of their choice $c_i \leftarrow E(k, m_{i,b})$, which is known as CPA. Additionally, an adversary can submit $i = [1, 2, \ldots, q]$ decryption queries of their choice $m_i \leftarrow D(k, c_i)$, where c_i submitted for decryption is not equal to c_i received from encryption queries. This is referred to as CCA. The advantage associated with the fact that the adversary can distinguish between $m_{i,0}$ and $m_{i,1}$ is negligible.

Theorem 1: The SE-Enc scheme is indistinguishable under IND-CCA3 such that for any q-query, there are two efficient adversaries B1 and B2 that satisfy equation 4.

$$Adv_{IND-CCA3}[A, E] \leq q * (Adv_{CPA}[B_2, E] + Adv_{CI}[B_1, E]) \quad (4)$$

Algorithm 9 Converting Binary Values to Plaintext Algorithm

```
Input: Binary values
Output: The plain text message $M$
1 Recipient : obtain the binary values;
2 Recipient :
  convert each 8bits into its corresponding char;
3 Recipient : repeat step 3 for all blocks;
4 Recipient : for each $N$ char aggregate to single block;
5 The message $M \leftarrow$ the results;
```

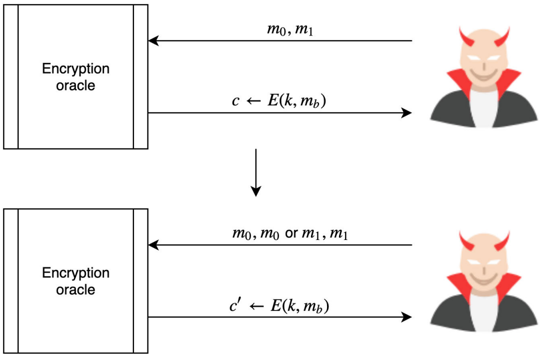

FIGURE 14. Adversary steps to perform CPA.

Proof: The proof of this theorem is divided into two parts. First, a proof is presented to show that the left part, $Adv_{CPA}[B_2, E]$, is negligible by the definition of IND-CPA, which is, $Adv_{CPA}[A, E] = |Pr[EXP(0) = 1] - Pr[EXP(0) = 1]|$, where the adversary only has access to the encryption oracle. When the adversary submits two messages ($m_{i,0}$ and $m_{i,1}$), the output of the first experiment is $c_{i,b} \leftarrow E(k, m_{i,b})$. The adversary can submit either ($m_{i,0}$ and $m_{i,0}$) or ($m_{i,1}$ and $m_{i,1}$) and the encryption oracle returns the $c'_{i,b} \leftarrow E(k, m_{i,b})$. Consequently, the adversary can match the results and determine the value of $b = 0, 1$ as depicted in Figure 14. However, in the SE-Enc scheme, a random IV is used for each encryption session, $c_{i,b} \leftarrow< IV$, $E(k, IV \oplus m_{i,b}) >$. Therefore, each time the adversary submits the same messages ($m_{i,j}$ and $m_{i,j}$) the encryption oracle will return $c_{i,j} \neq c_{i+1,j}$.

The proof of the right part, $Adv_{CI}[B_1, E]$, which is the adversary advantage of wining challenging game with Ciphertext Integrity (CI) begins by noting that the adversary has the power to access both the decryption oracle and the encryption oracle where the c_i submitted for decryption is not equal to the c_i received from encryption. Therefore, likewise the CCA the adversary can modify the received $c_i \leftarrow< IV$, $E(k, m_i) >$ by XORing the IV with random R, thereby guessing m_i. This process is illustrated in Figure 15. However, as the Ciphertext Integrity maintained, the decryption oracle in SE-Enc scheme ignores any modified c'_i, meaning that the oracle returns $\perp \leftarrow< IV', D(k, m_i) >$ for each modified c'_i. As a result, the $Adv_{CI}[B_1, E]$ is negligible, thus the $Adv_{IND-CCA3}[A, E]$ is negligible and this represents a proof that the SE-Enc scheme is indistinguishable under IND-CCA3. □

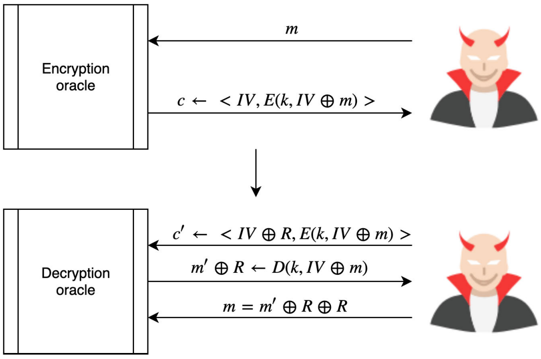
Encryption oracle
m
$c \leftarrow < IV, E(k, IV \oplus m) >$
Decryption oracle
$c' \leftarrow < IV \oplus R, E(k, IV \oplus m) >$
$m' \oplus R \leftarrow D(k, IV \oplus m)$
$m = m' \oplus R \oplus R$

FIGURE 15. Adversary steps to perform CCA.

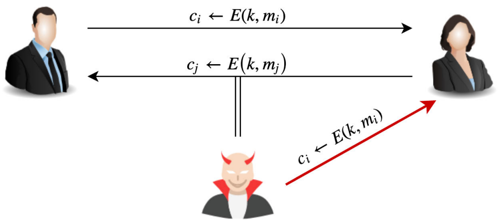
$c_i \leftarrow E(k, m_i)$
$c_j \leftarrow E(k, m_j)$
$c_i \leftarrow E(k, m_i)$

FIGURE 16. Adversary steps to perform replay attack.

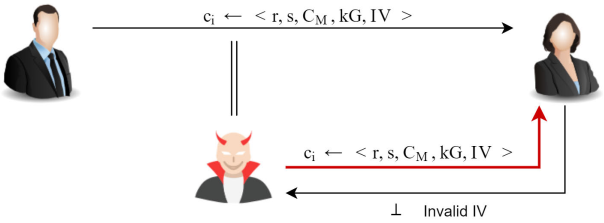
$c_i \leftarrow < r, s, C_M, kG, IV >$
$c_i \leftarrow < r, s, C_M, kG, IV >$
$\perp$ Invalid IV

FIGURE 17. SE-Enc scheme is resistant to the replay attack.

B. REPLAY ATTACK

An adversary can eavesdrop on an encrypted session between the sender and the recipient. Following this, adversary can intercept the encrypted data and resend it again to the recipient, as shown in Figure 16.

Theorem 2: The SE-Enc scheme is resistant to the replay attack.

Proof: When the sender and recipient establish a communication session that an adversary eavesdrops upon, the adversary can intercept the encrypted messages between the two parties. However, the adversary needs to establish a new session to replay the intercepted messages. Therefore, new session parameters are generated that must be used with a new session, meaning that the intercepted messages are ignored by the recipient as the signature is not valid (i.e. captured packet IV not equal new session IV). A visual illustration of this proof is given in the following figure. □

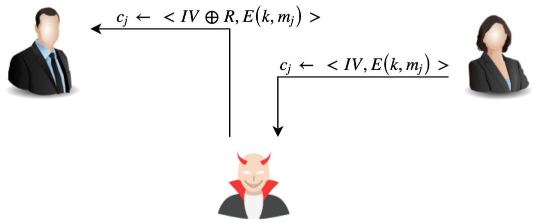
$c_j \leftarrow\ < IV \oplus R, E(k, m_j) >$
$c_j \leftarrow\ < IV, E(k, m_j) >$

FIGURE 18. Adversary modifies the IV to perform a malleability attack.

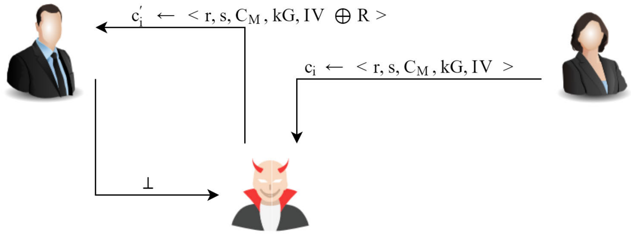
$c_i' \leftarrow\ < r, s, C_M, kG, IV \oplus R >$
$c_i \leftarrow\ < r, s, C_M, kG, IV >$
$\perp$

FIGURE 19. SE-Enc scheme is resistant to the malleability attack.

TABLE 2. Security comparison of SE-Enc and other schemes.

	1	2	3	4	5	6	7	8
Tiwari et al. [27]	Y	Y	Y	N	N	N	N	N
Singh et al. [28]	Y	Y	N	N	N	N	N	N
Sengupta et al. [29]	Y	N	N	N	N	N	N	N
Singh et al. [30]	Y	N	N	N	N	N	N	N
King et al. [32]	Y	Y	Y	N	N	N	N	N
SE-Enc	Y	Y	Y	Y	Y	Y	Y	Y

C. MALLEABILITY ATTACK

When encrypted data is modified, thus resulting in a modification to the decrypted message, this is referred to as a malleability attack. An adversary can XOR the IV with any value, the result being that the message is XORed with the same value. Figure 18 illustrates the steps involved in a malleability attack.

Theorem 3: The SE-Enc scheme is resistant to the malleability attack.

Proof: The adversary can intercept the encrypted messages transmitted between the two parties. In addition, they can modify the intercepted message and resend it to the recipient. However, the recipient will ignore the received message as the signature is not valid. A depiction of this proof is given in the following figure. □

Table 2 presents a comparison of the security analysis between SE-Enc and other existing schemes.

D. PERFORMANCE EVALUATION

The main objective of the SE-Enc scheme is to offer an AE scheme that provides an acceptable level of performance. The first part of this objective has been achieved, thereby necessitating a performance evaluation to compare the scheme with other schemes.

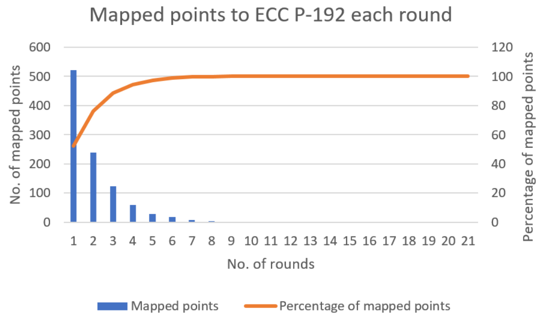

FIGURE 20. Experiment 1 - Number of rounds to map all random keys to EC secp192k1.

1) REDUCTION OF PADDING SIZE IN EACH BLOCK

Mapping any point to an elliptic curve must be undertaken correctly. Some points x_i cannot be pointed to an elliptic curve because there is no corresponding y_i that satisfies the elliptic curve's equation. Therefore, we increment x_i by 1 and try again. If there is a match, then the point is mapped; however, if there is no match, we increment the x_i again by 1 and try again. In many schemes, the numerical values that are mapped to the elliptic curve are obtained from the ASCII table by converting the text to a sequence of binary values. Thus, if these values are incremented, then the original value is changed. For instance, the numerical value of "H" based on the ASCII table is 72. If there is no match in the elliptic curve equation, this becomes 73 through incrementation, which represents the value "I". Prior to the mapping process, many schemes pad the value to create an 8-bit value, which ensures that the mapping is found and the original value remains unchanged.

3-bit padding is used in the SE-Enc scheme. Three experiments were conducted on three well-known elliptic curves, namely, secp192k1, NIST-224, and secp256k1. In each experiment, 1000 random numbers were generated for the 192-bit, 224-bit, and 256-bit sizes, and an attempt was made to map these to the corresponding elliptic curve. We found that approximately 50% of the values were mapped from the first round. Similarly, 75% of the values were mapped from the first and second rounds, and the values increased to 88% in the third round. Over, 98% of the values were mapped to the elliptic curve by the sixth round. In fact, for a 192-bit key, all values were mapped by the seventh round, thus the need for padding bits equal to 3, which allowed us to try up to eight rounds ($2^3 = 8\ rounds$). Figures 20, 21, and 22 show the experimental results.

The findings revealed that the SE-Enc scheme reduces the size of the padding bits to 3-bit. Thus, in comparison to other schemes, SE-Enc decreased the size of the padding bits for each block by over 65%. Figure 23 presents the results of a comparison between SE-Enc and other schemes.

2) ENCODING AND DECODING OPERATIONS

The second issue addressed in the performance evaluation of the SE-Enc scheme was the matter of encoding and decoding

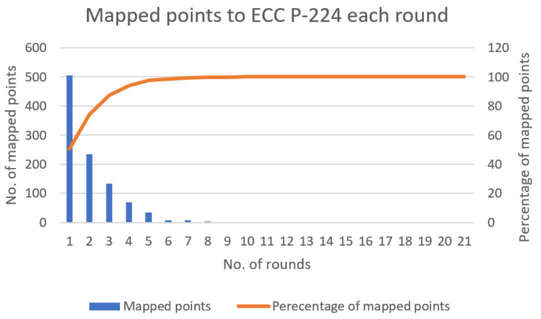

FIGURE 21. Experiment 2 - Number of rounds to map all random keys to EC NIST-224.

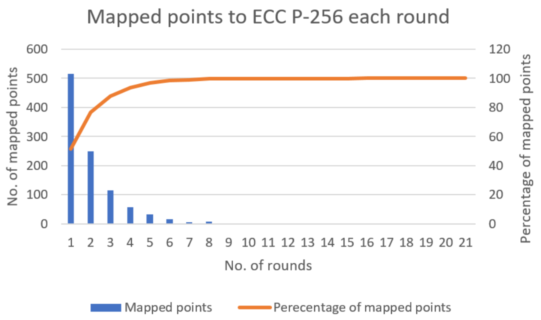

FIGURE 22. Experiment 3 - Number of rounds to map all random keys to EC secp256k1.

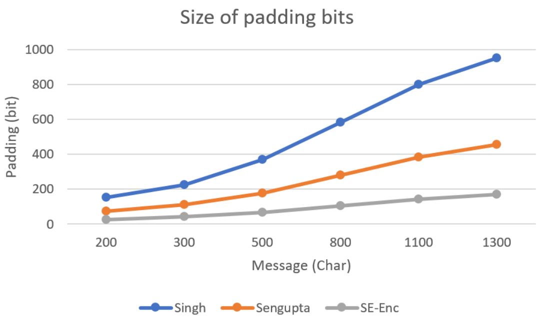

FIGURE 23. Size of padding bits for SE-Enc and other schemes.

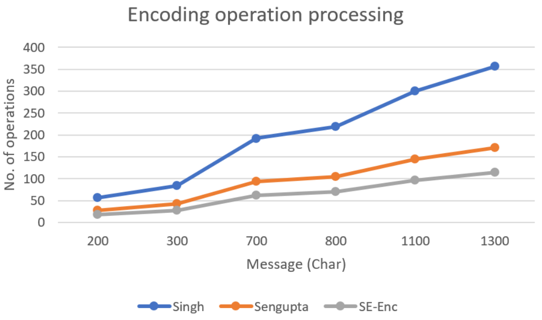

FIGURE 24. Number of operations required to encode a message.

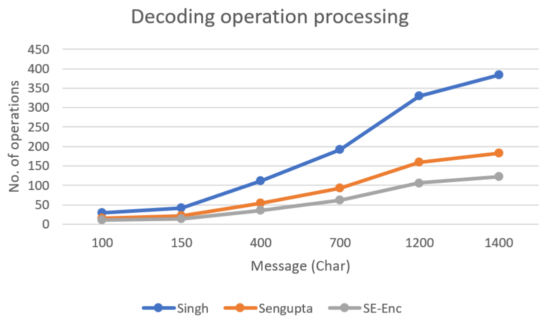

FIGURE 25. Number of operations required to decode a message.

operations. For each operation, several steps are required to perform it (e.g., converting characters to ASCII, converting ASCII to binary, and converting binary values to decimal values). The SE-Enc scheme performs two operations per block, while other schemes perform three operations per block. Figures 24 and 25 present the results for the comparison of the encoding and decoding operations between SE-Enc and other existing schemes.

VI. CONCLUSION AND FUTURE RESEARCH

This work introduced the SE-Enc scheme, an AE scheme based on the ECC that effectively encodes messages and maps them to an elliptic curve. Many schemes that ignore the encoding phase become vulnerable to encryption flaws. Therefore, the proposed scheme addresses the encoding phase and, as such, benefits from being resistant to several encryption attacks, including KPA, CPA, CCA, and several other active attacks. Moreover, a proof-based security analysis is conducted in this work to illuminate the degree to which the proposed scheme is secure. The padding reduction properties of the scheme are being discussed, along with the issue of several key-sizes based on well known and approved elliptic curves. Using variety of metrics, this work demonstrates that the SE-Enc scheme outperforms other techniques in terms of being resistance to attacks, padding sizes, the number of encoding operations, and the number of decoding operations.

In future research, the authors intend to study the implications of addressing the encryption property at the mapping phase instead of the encoding phase. In addition, security analysis and performance evaluation of the new scheme will be benchmarked against the SE-Enc scheme.

REFERENCES

[1] J. Guziur, M. Pawlak, and A. Poniszewska-Marańda, and B. Wieczorek, "Light blockchain communication protocol for secure data transfer integrity," in *Proc. Int. Symp. Cyberspace Saf. Secur.* Cham, Switzerland: Springer, 2018, pp. 194–208.

[2] M. La Torre, J. Dumay, and M. A. Rea, "Breaching intellectual capital: Critical reflections on big data security," *Meditari Accountancy Res.*, vol. 26, no. 3, pp. 463–482, 2018.

[3] S. W. Pritchard, G. P. Hancke, and A. M. Abu-Mahfouz, "Cryptography methods for software-defined wireless sensor networks," in *Proc. IEEE 27th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2018, pp. 1257–1262.

[4] G. Verma, M. Liao, D. Lu, W. He, X. Peng, and A. Sinha, "An optical asymmetric encryption scheme with biometric keys," *Opt. Lasers Eng.*, vol. 116, pp. 32–40, May 2019.

[5] H. N. Almajed, A. S. Almogren, and A. Altameem, "A resilient smart body sensor network through pyramid interconnection," *IEEE Access*, vol. 7, pp. 51039–51046, 2019.

[6] K. Haseeb, N. Islam, A. Almogren, I. U. Din, H. N. Almajed, and N. Guizani, "Secret sharing-based energy-aware and multi-hop routing protocol for IoT based WSNs," *IEEE Access*, vol. 7, pp. 79980–79988, 2019.

[7] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. London, U.K.: Springer, 2009.

[8] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.

[9] M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "Low-cost security of IoT sensor nodes with rakeness-based compressed sensing: Statistical and known-plaintext attacks," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 2, pp. 327–340, Feb. 2018.

[10] M. Liao, W. He, D. Lu, J. Wu, and X. Peng, "Security enhancement of the phase-shifting interferometry-based cryptosystem by independent random phase modulation in each exposure," *Opt. Lasers Eng.*, vol. 89, pp. 34–39, Feb. 2017.

[11] S. Ahmed, A. Zaman, Z. Zhang, K. M. R. Alam, and Y. Morimoto, "Semi-order preserving encryption technique for numeric database," *Int. J. Netw. Comput.*, vol. 9, no. 1, pp. 111–129, 2019.

[12] L. Davoli, L. Veltri, G. Ferrari, and U. Amadei, "Internet of things on power line communications: An experimental performance analysis," in *Smart Grids and Their Communication Systems*. Singapore: Springer, 2019, pp. 465–498.

[13] S. Debnath, M. V. Nunsanga, and B. Bhuyan, "Study and scope of signcryption for cloud data access control," in *Advances in Computer, Communication and Control*. Singapore: Springer, 2019, pp. 113–126.

[14] C. Boyd, B. Hale, S. F. Mjølsnes, and D. Stebila, "From stateless to stateful: Generic authentication and authenticated encryption constructions with application to TLS," in *Proc. Cryptographers' Track RSA Conf.* Cham, Switzerland: Springer, 2016, pp. 55–71.

[15] R. Zuccherato, "Elliptic curve cryptography support in entrust," Entrust Datacard, Ottawa, ON, Canada, Tech. Re. 1.0, May 2000.

[16] M. Tyagi, M. Manoria, and B. Mishra, "A framework for data storage security with efficient computing in cloud," in *Proc. Int. Conf. Adv. Comput. Netw. Inform.* Springer, 2019, pp. 109–116.

[17] J. Louw, G. Niezen, T. D. Ramotsoela, and A. M. Abu-Mahfouz, "A key distribution scheme using elliptic curve cryptography in wireless sensor networks," in *Proc. IEEE 14th Int. Conf. Ind. Inform. (INDIN)*, Jul. 2016, pp. 1166–1170.

[18] G. Kanda, A. O. Antwi, and K. Ryoo, "Hardware architecture design of aes cryptosystem with 163-bit elliptic curve," in *Advanced Multimedia and Ubiquitous Engineering*. Singapore: Springer, 2018, pp. 423–429.

[19] Z. E. Dawahdeh, S. N. Yaakob, and R. R. B. Othman, "A new modification for menezes-vanstone elliptic curve cryptosystem," *J. Theor. Appl. Inf. Technol.*, vol. 85, no. 3, p. 290, 2016.

[20] L. Ferretti, M. Marchetti, and M. Colajanni, "Fog-based secure communications for low-power IoT devices," *ACM Trans. Internet Technol.*, vol. 19, no. 2, p. 27, 2019.

[21] F. Albalas, M. Al-Soud, O. Almomani, and A. Almomani, "Security-aware coap application layer protocol for the Internet of things using elliptic-curve cryptography," *Power (mw)*, vol. 15, no. 3A, p. 151, 2018.

[22] S. Khan and R. Khan, "Elgamal elliptic curve based secure communication architecture for microgrids," *Energies*, vol. 11, no. 4, p. 759, 2018.

[23] A. U. Ay and C. Mancillas-López, E. Öztürk, F. Rodríguez-Henríquez, and E. Savaş, "Constant-time hardware computation of elliptic curve scalar multiplication around the 128 bit security level," *Microprocessors Microsyst.*, vol. 62, pp. 79–90, Oct. 2018.

[24] S. Liu, H. Yao, and X. A. Wang, "Fast elliptic curve scalar multiplication for resisting against spa," *Int. J. Comput. Sci. Eng.*, vol. 17, no. 3, pp. 343–352, 2018.

[25] S. Ezzouak and A. Azizi, "On the efficiency of scalar multiplication on the elliptic curves," in *Proc. Int. Conf. Eur. Middle East North Afr. Inf. Syst. Technol. Support Learn.* Cham, Switzerland: Springer, 2018, pp. 393–399.

[26] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[27] H. D. Tiwari and J. H. Kim, "Novel method for DNA-based elliptic curve cryptography for IoT devices," *ETRI J.*, vol. 40, no. 3, pp. 396–409, 2018.

[28] L. D. Singh and K. M. Singh, "Image encryption using elliptic curve cryptography," *Procedia Comput. Sci.*, vol. 54, pp. 472–481, Aug. 2015.

[29] A. Sengupta and U. K. Ray, "Message mapping and reverse mapping in elliptic curve cryptosystem," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5363–5375, 2016.

[30] L. D. Singh and K. M. Singh, "Implementation of text encryption using elliptic curve cryptography," *Procedia Comput. Sci.*, vol. 54, pp. 73–82, Aug. 2015.

[31] P. Das and C. Giri, "An efficient method for text encryption using elliptic curve cryptography," in *Proc. IEEE 8th Int. Adv. Comput. Conf. (IACC)*, Dec. 2018, pp. 96–101.

[32] B. King, "Mapping an arbitrary message to an elliptic curve when defined over GF (2^n)," *Int. J. Netw. Secur.*, vol. 8, no. 2, pp. 169–176, 2009.

[33] H. N. AlMajed and A. S. AlMogren, "Simple and effective secure group communications in dynamic wireless sensor networks," *Sensors*, vol. 19, no. 8, p. 1909, 2019.

[34] Y. Yin, L. Wu, Q. Peng, and X. Zhang, "A novel SPA on ECC with modular subtraction," in *Proc. 12th IEEE Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Nov. 2018, pp. 179–182.

[35] S. D. Galbraith and F. Vercauteren, "Computational problems in supersingular elliptic curve isogenies," *Quantum Inf. Process.*, vol. 17, no. 10, p. 265, 2018.

[36] T. Wu and R. Wang, "Fast unified elliptic curve point multiplication for NIST prime curves on FPGAs," *J. Cryptograph. Eng.*, pp. 1–10, 2019.

[37] T. Shahroodi, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, "Low-latency double point multiplication architecture using differential addition chain over GF(2^m)," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 4, pp. 1465–1473, Apr. 2019.

[38] A. Mrabet, N. El-Mrabet, R. Lashermes, J.-B. Rigaud, B. Bouallegue, S. Mesnager, and M. Machhout, "High-performance elliptic curve cryptography by using the CIOS method for modular multiplication," in *Proc. Int. Conf. Risks Secur. Internet Syst.* Cham, Switzerland: Springer, 2016, pp. 185–198.

[39] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Trans. Dependable Secure Comput.*, to be published.

[40] M. G. G. Ganesh, "Secure method for text encryption using elliptic curve cryptography," *Int. J.*, vol. 3, no. 11, pp. 11–15, 2018.

[41] D. Mahto, "Data communication security modeling using elliptic curve cryptography and biometrics," Ph.D. dissertation, Dept. Comput. Appl., NIT Jamshedpur, India, 2018.

[42] R. Kumar, "Cryptanalysis of protocol for enhanced threshold proxy signature scheme based on elliptic curve cryptography for known signers," in *Knowledge Computing and Its Applications*. Singapore: Springer, 2018, pp. 191–211.

[43] Z. Liu and H. Seo, "Iot-nums: Evaluating nums elliptic curve cryptography for iot platforms," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 720–729, Mar. 2019.

[44] D. P. Shah and P. G. Shah, "Revisting of elliptical curve cryptography for securing Internet of Things (IOT)," in *Proc. Adv. Sci. Eng. Technol. Int. Conf. (ASET)*, Feb./Apr. 2018, pp. 1–3.

[45] A. P. Fournaris, C. Dimopoulos, A. Moschos, and O. Koufopavlou, "Design and leakage assessment of side channel attack resistant binary edwards elliptic curve digital signature algorithm architectures," *Microprocessors Microsyst.*, vol. 64, pp. 73–87, Feb. 2019.

[46] A. G. Reddy, A. K. Das, V. Odelu, A. Ahmad, and J. S. Shin, "A privacy preserving three-factor authenticated key agreement protocol for client–server environment," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 2, pp. 661–680, 2019.

[47] S. N. AlSaad and A. K. Naji, "Elliptic curve video encryption in mobile phone based on multi-keys and chaotic map," *Al-Mustansiriyah J. Sci.*, vol. 29, no. 2, pp. 106–116, 2018.

[48] O. Reyad, "Text message encoding based on elliptic curve cryptography and a mapping methodology," *Inf. Sci. Lett*, vol. 7, no. 1, pp. 7–11, 2018.

[49] K. Keerthi and B. Surendiran, "Elliptic curve cryptography for secured text encryption," in *Proc. Int. Conf. Circuit ,Power Comput. Technol. (ICCPCT)*, Apr. 2017, pp. 1–5.

[50] F. Amounas and E. El Kinani, "Fast mapping method based on matrix approach for elliptic curve cryptography," *Int. J. Inf. Netw. Secur.*, vol. 1, no. 2, pp. 54–59, 2012.

[51] J. Muthukuru and B. Sathyanarayana, "Fixed and variable size text based message mapping technique using ECC," *Global J. Comput. Sci. Technol.*, to be published.

[52] B. Padma, D. Chandravathi, and R. P. Prapoorna, "Encoding and decoding of a message in the implementation of Elliptic Curve cryptography using Koblitz's method," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 5, pp. 1904–1907, Aug. 2010.

[53] D. P. Shah and N. P. Shah, "Implementation of digital signature algorithm by using elliptical curve p-192," *Austral. J. Wireless Technol., Mobility Secur.*, vol. 1, no. 1, pp. 1–4, 2019.

[54] K. E. Abdullah and N. H. M. Ali, "Security improvement in elliptic curve cryptography," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 5, pp. 122–131, 2018.

[55] X. Hu, X. Zheng, S. Zhang, W. Li, S. Cai, and X. Xiong, "A high-performance elliptic curve cryptographic processor of SM2 over GF($_p$)," *Electron.*, vol. 8, no. 4, p. 431, 2019.

[56] K. K. F. Yuen, "Towards a cybersecurity investment assessment method using primitive cognitive network process," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2019, pp. 068–071.

[57] C. Biswas, U. D. Gupta, and M. M. Haque, "An efficient algorithm for confidentiality, integrity and authentication using hybrid cryptography and steganography," in *Proc. Int. Conf. Elect., Comput. Commun. Eng. (ECCE)*, Feb. 2019, pp. 1–5.

[58] R. T. Tiburski, C. R. Moratelli, S. F. Johann, M. V. Neves, E. de Matos, L. A. Amaral, and F. Hessel, "Lightweight security architecture based on embedded virtualization and trust mechanisms for IoT edge devices," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 67–73, Feb. 2019.

[59] S. Kanchan and N. S. Chaudhari, "Signcrypting the group signature with non-transitive proxy re-encryption in VANET," in *Recent Findings in Intelligent Computing Techniques*. Singapore: Springer, 2019, pp. 15–23.

[60] F.-L. Chen, Z.-H. Wang, and Y.-M. Hu, "A new quantum blind signature scheme with BB84-state," *Entropy*, vol. 21, no. 4, p. 336, 2019.

[61] Y. Zhou, Z. Li, F. Hu, and F. Li, "Identity-based combined public key schemes for signature, encryption, and signcryption," in *Information Technology and Applied Mathematics*. Singapore: Springer, 2019, pp. 3–22.

[62] A. S. Kittur and A. R. Pais, "A trust model based batch verification of digital signatures in iot," *J. Ambient Intell. Hum. Comput.*, to be published.

[63] A. I. Gomez, D. Gomez-Perez, and G. Renault, "A probabilistic analysis on a lattice attack against dsa," *Des., Codes Cryptogr.*, to be published.

[64] A. C. Aldaya, B. B. Brumley, A. J. C. Sarmiento, and S. Sánchez-Solano, "Memory tampering attack on binary gcd based inversion algorithms," *Int. J. Parallel Program.*, to be published.

HISHAM N. ALMAJED received the bachelor's degree in information systems from the College of Computer and Information Sciences, King Saud University, in 2004, and the M.Sc. degree in computer applications and systems administration from the Computer Section, Arab East colleges, in 2015. He is currently pursuing the Ph.D. degree in computer science from the College of Computer and Information Sciences, King Saud University. Presently, he is working at Saline Water Conversion Corporation head quarter in Riyadh, Saudi Arabia, as an information technology governance team member. He received several professional certifications, including PMP, CISA, ISO27001 Lead Auditor, ISO27001 Leas Implementer, TOGA9, and ITIL Expert. His research interests include computer security and wireless sensor network security.

AHMAD S. ALMOGREN received the Ph.D. degree in computer sciences from Southern Methodist University, Dallas, TX, USA, in 2002. He was an Assistant Professor of computer science and a member of the Scientific Council with the Riyadh College of Technology. He was also the Dean of the College of Computer and Information Sciences and the Head of the Council of Academic Accreditation with Al Yamamah University. He is currently an Associate Professor and the Vice Dean for the development and quality with the College of Computer and Information Sciences, King Saud University, Saudi Arabia. His research interests include mobile and pervasive computing, computer security, sensor and cognitive networks, and data consistency. He has served as a Guest Editor for several computer journals.