Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7499836, 11 pages
https://doi.org/10.1155/2022/7499836

Research Article

An Elliptic Curve Signcryption Scheme and Its Application

Ping Zhang, Yamin Li, and Huanhuan Chi

School of Mathematics and Statistics, Henan University of Science and Technology, China

Correspondence should be addressed to Yamin Li; 2390043823@qq.com

Received 24 November 2021; Revised 21 February 2022; Accepted 8 April 2022; Published 6 May 2022

Academic Editor: Mu Zhou

Two basic security requirements in communication are confidentiality and authentication. Signcryption is an ideal technique to transmit encrypted and authenticated data. In view of the shortcomings of existing signcryption schemes and the high security of elliptic curve cryptography (ECC), we design a ECC-based signcryption scheme and evaluate it in terms of security, computational overhead, and communication overhead. Finally, we consider the application of our secure and efficient signcryption scheme in the smart lock key management system and analyze the bit-oriented performance of the designed key management scheme.

1. Introduction

With the rapid development of Internet, there are an increasing number of smart devices, among which the smart lock is one of the typical representatives. Compared with other smart devices, the smart lock requires higher security. When designing the smart lock, the security is the first problem to be considered.

Confidentiality and authentication are two basic security requirements in communication. In general, encryption can ensure the confidentiality of the message, and digital signature can ensure the authentication of the message. In order to meet these two requirements at the same time, the traditional method is either "Encrypt before signing" or "sign before encryption". However, these will result in a large amount of computation and communication costs. In 1997, Zheng [1] firstly proposed the notion of signcryption. Signcryption not only meets these two security requirements at the same time, but also its computational and communication costs are much lower than the traditional methods described above. Signcryption is an ideal way to transmit information encrypted and authenticated. Therefore, it also can be used for mobile device authentication. The information on which authentication is based generally includes the following three categories: (1) information known to the user, such as passwords; (2) things owned by the user, such as smart cards; and (3) biometrics of the user, such as fingerprints. Single-factor authentication generally refers to password-based authentication. Two-factor authentication refers to the smart-card-based password authentication. Multifactor authentication refers to authentication that uses two or more pieces of information. Signcryption has broad application prospects in e-commerce, e-government, and key management.

At present, the secure and practical public key cryptosystems include RSA cryptosystem (based on the big integer factorization problem), DSA cryptosystem (based on the discrete logarithm problem in the finite field), and ECC cryptosystem (based on the ECDLP). Among them, ECC cryptosystem has the highest security when the key length is the same.

The ECC cryptosystem was independently proposed by Neal Koblitz [2] and V. S. Miller [3] in 1985. It uses the elliptic curve whose variables and coefficients are elements in the finite field. The security of ECC is based on the ECDLP. Different from the discrete logarithm problem in the finite multiplication group, the ECDLP on the finite field is more difficult to solve, which cannot be solved by all known algorithms in polynomial time. In the general discrete logarithm problem, the algebraic operation on the finite field includes two operations, field addition and field multiplication, which makes the general discrete logarithm problem can be solved in subexponential time. However, in the ECDLP, the algebraic operation only includes the point addition operation

on the elliptic curve. Therefore, all the discrete logarithm algorithms cannot solve the ECDLP in subexponential time except some very special elliptic curves.

In view of the shortcomings of the existing key management scheme of the smart lock, the advantages of signcryption scheme, and the high security of ECC, this paper designs a signcryption scheme based on elliptic curve and firstly applies the signcryption scheme to the key management scheme of the smart lock system.

1.1. Related Works. Since the signcryption scheme was put forward in 1997, there have been several specific schemes based on different difficult assumptions ([1, 4–6]). In addition to the basic security objectives, some new features are introduced in the study of signcryption schemes, such as identity-based signcryption scheme ([6–11]), hybrid signcryption scheme [12], key encapsulation mechanism (KEM) and data encapsulation mechanism (DEM-)-based signcryption scheme [13], certificateless signcryption scheme [14], verifiable signcryption scheme [10], attribute-based signcryption scheme ([15, 16]), functional signcryption scheme [17], or key invisible signcryption scheme [18].

Malone-Lee [7] defined the security model of identity-based signcryption scheme in 2002 and constructed the first identity-based signcryption scheme using bilinear pairings. In 2003, Nalla et al. [19] proposed an identity-based signcryption scheme on bilinear pairings of elliptic curves. This scheme is an improvement of Lee's [7] signcryption scheme. In 2004, with the difficulty of q-Diffie-Hellman problem (q-DH) in Gap-Diffie-Hellman group, Libert et al. [20] proposed a new public key authenticated signcryption scheme. This scheme is particularly efficient. The cost of signcryption operation is almost the same as that of ElGamal encryption, and the inverse operation only needs one pairing evaluation and three power calculations. Under the assumption of q-strong Diffie-Hellman, they proved the unforgeability of this scheme. In 2009, based on the encryption scheme of water [21], Yu et al. [22] proposed the first identity-based signcryption scheme without random oracle.

In 2012, Kar [23] proposed a provably secure signcryption scheme in the random oracle model by modifying the scheme of Libert et al. [24]. This scheme is safer and more reliable than the scheme of Libert et al. In the random oracle model, they use two hypotheses, strong Diffie-Hellman (SDH) and Diffie-Hellman inversion (DHI), to prove the security of the scheme. In the same year, S. Sharmila et al. [11] firstly proposed an identity-based signcryption scheme with provable security under the standard model. The unforgeability of the scheme is based on the difficulty of computational Diffie-Hellman problem (CDH), and the indistinguishability is based on the difficulty of decisional bilinear Diffie-Hellman problem (DBDH). In 2013, Kar [25] proposed an aggregate signcryption scheme with provable security. The security of the scheme is based on the computational reliability of DBDH and discrete logarithm problem (DL). In 2014, Liu Zhenhua et al. [26] proposed a new revocable identity-based signcryption scheme to revoke malicious users in the signcryption system. In this scheme, the master key is randomly divided into two parts, one is used to construct the initial key, and the other is used to update the key. In the standard model, they proved the IND-CCA2 security based on DBDH difficult problem and the EUF-CMA security based on CDH difficult problem. In 2015, Braeken et al. [27] pointed out some problems of existing pairing-free signcryption scheme. Then, they modified the scheme and extended it to a multiuser signcryption scheme. In 2016, Kar and Naik [28] proposed an effective certificateless signcryption scheme based on bilinear mapping in the random oracle model. They proved the security of the scheme based on the assumptions of the k-CAA, Inv-CDH, q-BDHI, and CDH. In the same year, Han Yiliang et al. [29] combined Niedereiter public key cryptography with CFS signature scheme and constructed a signcryption scheme. This scheme can resist quantum attack and has a small amount of key data. They proved the IND-CCA2 security and EUF-CMA security of the scheme in the random oracle model. In 2017, Zhou Yanwei et al. [30] proposed an efficient certificateless signcryption scheme without bilinear mapping and proved the security of the scheme based on CDH and DL in the random oracle model. Tsai et al. [31] proposed a new multidocument blind signature scheme based on ECC. This scheme adds the design of the signature encryption paradigm to the blind signature scheme to enhance high-level security. In 2018, for the security of hybrid signcryption schemes, Dai et al. [32] studied the replayable CCA security (RCCA) of SKEM+DEM [33] and Tag SKEM+DEM [13]. If the scheme SKEM is RCCA secure and the scheme DEM is RCCA secure, the hybrid signature scheme SKEM+DEM is RCCA secure. If the scheme Tag-SKEM is RCCA secure and the scheme DEM is RCCA secure, the Tag SKEM + DEM hybrid encryption scheme is RCCA secure. In the single-factor authentication research area, He Debiao et al. [34] proposed a password-based remote user authentication scheme without smart cards. The scheme can resist various attacks, such as device stolen attack and privileged insider attack. In the two-factor authentication research area, Wang Ding et al. [35] proposed a smart-card-based password authentication scheme that kills two birds with one stone. By integrating "honeywords" with their proposed "fuzzy-verifiers," the scheme not only not only eliminates the long-standing security-usability conflict that is considered intractable in the literature, but also achieves security guarantees beyond the conventional optimal security bound. Our signcryption scheme has highly efficient and satisfies multiple security properties; we believe it can be used as a building block for the authentication phase of a single-factor authentication scheme. When the server and user authenticate each other and generate a session key, they can use our scheme to signcrypt their own messages, respectively, which not only achieves authentication but also provides additional confidentiality.

At present, there have been many works on the key management system of smart locks. For data security in narrow band Internet of things (NB-IOT) application environment, Jia Rongyuan et al. [36] proposed a lightweight encryption algorithm and encryption model based on AES [37] and chaos sequence. However, they did not explain how to transmit the key. There are problems such as difficult monitoring,

high power consumption requirements, and insecure wireless transmission of wireless smart lock. In order to solve the problems, Zhang Huanlan et al. [38] proposed a 433 MHz wireless module based on Diffie-Hellman key exchange algorithm and corrected block tiny encryption algorithm for double encryption smart lock system. In 2019, under the unreliable UDP data transmission of NB-IoT, Liu Mengjun [39] designed a key transmission interaction scheme to complete the reliable update of the user's unlock key with as little calculation and communication as possible. However, this scheme will continue to use the old key for unlocking when the unlock key update fails, which is not applicable to public rental housing. Because if the user loses the qualification to rent a house, the unlock key must be updated as soon as possible. In addition, in this work, the session key used between the server and the smart lock has low security. Sha Tao et al. [40] designed an identity verification mechanism based on position proof. They also proposed a timestamp encryption mechanism to prevent remote unlocking and replay attacks by malicious users. However, this work did not explain how the server issued the unlock key to the smart lock, and the smart lock did not upload operating information to the server. Wang et al. [41] designed a complementary multidimensional feature fusion network-based hand gesture recognition (CMFF-HGR) to extract features and achieve hand gesture recognition. The smart lock key management system based on hand gesture recognition is different from the key management scheme proposed in this paper. The smart lock system based on hand gesture recognition requires to memorize the gestures manually, and the hand gesture is easy to be known by others during the unlocking process. However, the key management scheme in this paper does not require manually memorizing the unlock key, and every time the unlock key is different, not being fixed. Therefore, the key management scheme in this paper has higher security.

1.2. Contribution. This paper proposed an efficient and secure ECC-based signcryption scheme and applied it to a smart lock key management system. To the best of our knowledge, this work is the first to consider the application of a signcryption scheme in a smart lock key management system. Compared with other smart lock key management schemes, our scheme is more efficient and secure due to the confidentiality and authentication by the signcryption itself, as well as the efficiency and other security properties of our signcryption scheme. In addition, in our key management scheme, the unlock key is delivered to the smartphone by the server, and then, the smartphone unlocks the smart lock through Bluetooth. Therefore, the unlock key is different every time, and the user does not need to memorize a fixed unlock key, which makes our key management scheme more secure and convenient.

1.3. Organization. This paper is organized as follows. The first section is the introduction of this paper. The second section introduces the basic knowledge, including elliptic curve discrete logarithm problem, the formal definition, and the security model of signcryption scheme. In the third section, we design a signcryption scheme based on elliptic curve and analyze the correctness, security, and performance of our signcryption scheme. In the fourth section, we apply our signcryption scheme to the key management system of smart lock. Finally, in the fifth section, we summarize the full text and give an outlook for future work.

2. Preliminaries

2.1. Basic Notation. In the following sections, if $|\mathrm{negl}(\lambda)| < 1/\mathrm{poly}(\lambda)$ for all polynomials $\mathrm{poly}(\lambda)$ and all sufficiently large λ, we call $\mathrm{negl}(\lambda)$ is negligible. In this paper, "PPT" represents probabilistic polynomial time.

2.2. Elliptic Curve Discrete Logarithm Problem

Definition 1 (Elliptic curve discrete logarithm problem). Given an elliptic curve $E(GF(q))$, P is a point on this elliptic curve and its order is a large prime number n $(ord(P) = n)$. For any random number d, $Q = dP$ can be easily calculated. However, if P and Q are known, it is very difficult to find d.

2.3. The Definition of Signcryption Scheme

2.3.1. Syntax. Given the key space $\mathcal{K}$, message space $\mathcal{M}$, and signcryption space $\mathcal{S}$, for any sender and receiver, a signcryption scheme SC = (setup, keygen, signcrypt, unsigncrypt) is a collection of the following four algorithms.

(i) Setup $(1^\lambda) \longrightarrow cp$: This is system initialization algorithm. This algorithm requires a security parameter λ as the input of the algorithm and requires common parameters cp as the output of the algorithm

(ii) Keygen $(cp, r) \longrightarrow (PK, SK)$: This is key generation algorithm, which is a random algorithm. This algorithm requires common parameters cp and random number r as the input of the algorithm and requires key pair (PK, SK) $(PK, SK \in \mathcal{K})$ as the output of the algorithm

(iii) Signcrypt $(cp, SK_S, PK_R, m) \longrightarrow \sigma$: This is signcryption algorithm. This algorithm requires common parameters cp, private key SK_S $(SK_S \in \mathcal{K})$ of sender, public key PK_R $(PK_R \in \mathcal{K})$ of receiver, and message m $(m \in \mathcal{M})$ as the input of the algorithm and requires signcryption $\sigma(\sigma \in \mathcal{S})$ as the output of the algorithm

(iv) Unsigncrypt $(cp, SK_R, PK_S, \sigma) \longrightarrow m$: This is unsigncryption algorithm. This algorithm requires common parameters cp, private key SK_R $(SK_R \in \mathcal{K}$) of receiver, and public key PK_S $(PK_S \in \mathcal{K})$ of sender and signcryption $\sigma(\sigma \in \mathcal{S})$ as the input of the algorithm. This algorithm outputs message m $(m \in \mathcal{M})$ or symbol "$\perp$" ("$\perp$" indicates that the unsigncryption failed)

Definition 2 (Correctness). For any message $m \in \mathcal{M}$, any sender (his key pair (SK_S, PK_S) was generated by $Keygen(c$

p, r)), any receiver (his key pair (SK_R, PK_R) was generated by $Keygen(cp, r)$), and the following formula holds

$$\text{Unsigncrypt}(cp, SK_R, PK_S, \text{Signcrypt}(cp, SK_S, PK_R, m)) = m. \quad (1)$$

2.4. The Security Model of Signcryption Scheme

Definition 3 (Confidentiality). The confidentiality security can be seen as a game between the adversary $\mathcal{O}$ and the challenger $\mathcal{C}$. This game is divided into five phases.

(i) Keygen phase: Challenger $\mathcal{C}$ runs algorithm Keygen(cp, r) to generate a sender key pair (SK_S, PK_S) and a receiver key pair (SK_R, PK_R), and sends (PK_S, PK_R) to adversary $\mathcal{O}$

(ii) Query phase 1: The adversary $\mathcal{O}$ sends multiple signcryption queries and unsigncryption queries to the challenger $\mathcal{C}$

(1) Signcryption query: The adversary $\mathcal{O}$ submits the message m and the public key (PK_S, PK_R) to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ calculates $\sigma = \text{signcrypt}(cp, SK_S, PK_R, m)$ and sends the result σ to the adversary $\mathcal{O}$

(2) Unsigncryption query: The adversary $\mathcal{O}$ submits the legitimate signcryption result σ and the public key (PK_S, PK_R) to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ calculates unsigncrypt (cp, SK_R, PK_S, σ) and sends the message m or symbol "$\perp$" to the adversary $\mathcal{O}$

(iii) Challenge phase: The adversary $\mathcal{O}$ submits two messages m_0, m_1 (m_0, m_1 have the same length) to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ randomly selects $i \in \{0, 1\}$, calculates $\sigma^* = \text{signcrypt}(cp, SK_S, PK_R, m_i)$ and sends the result σ^* to the adversary $\mathcal{O}$

(iv) Query phase 2: Similar to the query phase 1, the adversary $\mathcal{O}$ continues to send multiple signcryption queries and unsigncryption queries to the challenger $\mathcal{C}$ (the adversary $\mathcal{O}$ is forbidden from sending unsigncryption query for the result σ^*)

(v) Guess phase: The adversary $\mathcal{O}$ outputs a value i' as the guess for i. If $i' = i$, the adversary $\mathcal{O}$ wins this game

In this game, the advantage of the adversary $\mathcal{O}$ is $Adv(\mathcal{O}) = |\Pr[i' = i] - 1/2|$.

Definition 4 (Unforgeability). The unforgeability security can be seen as a game between the adversary $\mathcal{O}$ and the challenger $\mathcal{C}$. This game is divided into three phases.

(i) Keygen phase: Challenger $\mathcal{C}$ runs algorithm keygen (cp, r) to generate a sender key pair (SK_S, PK_S) and a receiver key pair (SK_R, PK_R) and sends (PK_S, PK_R) to adversary $\mathcal{O}$

(ii) Query phase: The adversary $\mathcal{O}$ sends multiple signcryption queries and unsigncryption queries to the challenger $\mathcal{C}$

(1) Signcryption query: The adversary $\mathcal{O}$ submits the message m and the public key (PK_S, PK_R) to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ calculates $\sigma = \text{signcrypt}(cp, SK_S, PK_R, m)$ and sends the result σ to the adversary $\mathcal{O}$

(2) Unsigncryption query: The adversary $\mathcal{O}$ submits the legitimate signcryption result σ and the public key (PK_S, PK_R) to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ calculates unsigncrypt (cp, SK_R, PK_S, σ) and sends the message m or symbol "$\perp$" to the adversary $\mathcal{O}$

(iii) Forgery phase: The adversary $\mathcal{O}$ submits the challenging content, including challenging message m^* and the forged signcryption σ^*. The challenger $\mathcal{C}$ submits the above input to the oracle, and the oracle returns the unsigncryption of signcryption σ^* to the challenger $\mathcal{C}$. If the result is message m^*, and the adversary $\mathcal{O}$ has not used this message as the input for signcryption query before, the adversary $\mathcal{O}$ wins this game

In this game, the advantage of the adversary is his probability of winning the game.

3. Our ECC-Based Signcryption Scheme

3.1. Construction. In this section, we define and construct our elliptic curve signcryption scheme SC = (setup, keygen, signcrypt, unsigncrypt).

(i) Setup: Let $GF(q)$ be a finite field of order q (the length of q is l), $E: y^2 = x^3 + ax + b (\text{mod } q)$ $(a, b \in GF(q), 4a^3 + 27b^2 \neq 0)$ be an elliptic curve in finite field $GF(q)$, P be the base point of the elliptic curve E. $ord(P) = n$, where n is a large prime number. Let $h = \#E(GF(q))/n$ $(h \ll n)$ is the cofactor. $\#E(GF(q))$ represents the number of points of the elliptic curve E defined on the finite field $GF(q)$). G_1 is an elliptic curve cyclic multiplication group of order q generated by point P. We suppose the plaintext space is $\{0, 1\}^l$ and select two hash functions $H_1: G_1 \longrightarrow \{0, 1\}^l$, $H_2: \{0, 1\}^* \longrightarrow Z_q$. Then, we expose parameters $D = \{q, l, a, b, P, G_1, n, h\}$ and hash function H_1, H_2

(ii) Keygen: The sender randomly selects SK_S as his private key, and his public key is $PK_S = SK_S P$. The receiver randomly selects SK_R as his private key, and his public key is $PK_R = SK_R P$. Then, they keep the private key SK_S, SK_R secret and expose the public key PK_S, PK_R

(iii) Signcrypt: The sender uses PK_R and SK_S to signcrypt message m

(a) Select a random number $k \in [1, n-1]$

(b) Compute $kPK_R = K$

(c) Compute $b = H_1(K)$.

(d) Compute $c = b \oplus m$

(e) Compute $e = H_2(m, K, PK_S, PK_R)$.

(f) Compute $s = k^{-1}(e + SK_S)$. If $s = 0$, return to step 1

(g) Get the signcryption $\sigma = (c, e, s)$, and send it to the recriver

(iv) Unsigncrypt: The receiver gets the signcryption $\sigma = (c, e, s)$ and uses PK_S and SK_R to unsigncrypt it

(a) Compute $w = s^{-1}$

(b) Compute $X = ewPK_R + wPK_SSK_R$

(c) Compute $b' = H_1(X)$.

(d) Compute $m = b' \oplus c$

(e) Compute $e' = H_2(m, X, PK_S, PK_R)$.

(f) If $e' = e$, return m, otherwise return "$\perp$".

3.2. Correctness. Because $s = k^{-1}(e + SK_S)$, we have $s^{-1} = k(e + SK_S)^{-1}$. Therefore, the following formula holds

$$\begin{aligned} X &= ewPK_R + wPK_SSK_R = es^{-1}PK_R + s^{-1}PK_SSK_R \\ &= es^{-1}SK_RP + s^{-1}SK_SSK_RP = (e + SK_S)s^{-1}SK_RP \\ &= (e + SK_S)k(e + SK_S)^{-1}SK_RP = kSK_RP = kPK_R = K. \end{aligned} \tag{2}$$

So, we have $b' = b$, $e' = e$. Here, $b' = b$ ensures that the receiver can restore the sender's message m; that is, the decryption process is correct. $e' = e$ ensures that the receiver can verify the correctness of the sender's signature; that is, the verification process is correct. Therefore, our signcryption scheme is correct.

3.3. Security

3.3.1. Confidentiality. Confidentiality means that information can only be used by authorized users and cannot be disclosed to unauthorized users. Confidentiality is a required property of encryption. Since signcryption needs to realize both signature and encryption, the signcryption scheme must also have confidentiality. According to Theorem 5, our signcryption scheme has confidentiality.

Theorem 5. *In the random oracle model, if there is an adversary $\mathcal{O}$ who can win the game of Definition 3 with the advantage of ε, there is a challenger $\mathcal{C}$ who can solve the ECDLP problem with the advantage of at least $\varepsilon' \geq \varepsilon/(q_{H_2} + q_{Sig} + q_{Uns})$. q_{H_2}, q_{Sig}, and q_{Uns} represent the number of times the adversary initiates H_2 query, signcryption query, and unsigncryption query, respectively.*

Proof. At the beginning of the game, the challenger $\mathcal{C}$ runs algorithm keygen (cp, r) to generate a sender key pair (SK_S, PK_S) and a receiver key pair (SK_R, PK_R) and sends (PK_S, PK_R) to adversary $\mathcal{O}$. The challenger $\mathcal{C}$ manages four lists $L_{H_1}, L_{H_2}, L_{Sig}, L_{Uns}$, which are initially empty. L_{H_1}, L_{H_2} are used to track the adversary's queries to oracle H_1, H_2, respectively, L_{Sig} is used to simulate signcryption oracle, and L_{Uns} is used to simulate unsigncryption oracle. □

Next, the adversary $\mathcal{O}$ sends queries to the challenger $\mathcal{C}$.

(1) (H_1 query) If (K, b) already exists in the list L_{H_1}, the challenger returns b. Otherwise, the challenger selects b from $\{0,1\}^l$ randomly, stores b in list L_{H_1}, and returns b

(2) (H_2 query) If (m, K, PK_S, PK_R, e) already exists in the list L_{H_2}, the challenger returns e. Otherwise, the challenger selects e from Z_q randomly, stores (m, K, PK_S, PK_R, e) in list L_{H_2}, and returns e

(3) (Signcrypt query) The public key of sender is PK_S, the public key of receiver is PK_R, and the message is m. The challenger selects k from $[1, n-1]$ randomly and computes $K = kPK_R$, $b = H_1(K)$. $H_1(K)$ can be obtained from the above H_1 query. Then, the challenger computes $c = b \oplus m$, $e = H_2(m, K, PK_S, PK_R)$. $e = H_2(m, K, PK_S, PK_R)$ can be obtained from the above H_2 query. The challenger computes $s = k^{-1}(e + SK_S)$ and returns (c, e, s)

(4) (Unsigncrypt query) The public key of sender is PK_S, the public key of receiver is PK_R, and the signcryption is $\sigma = (c, e, s)$. The challenger computes $w = s^{-1}$, $X = ewPK_R + wPK_SSK_R$. If $X \notin L_{H_1}$, the challenger returns "$\perp$", else computes $b' = H_1(X)$, $m = b' \oplus c$. If $(m, X, PK_S, PK_R) \notin L_{H_2}$, the challenger returns "$\perp$", else computes $e' = H_2(m, X, PK_S, PK_R)$. If $e' \neq e$, the challenger returns "$\perp$", else computes m

After the above-mentioned queries are initiated polynomial times, the game enters the challenge phase. The adversary $\mathcal{O}$ outputs two messages $\{m_0, m_1\}$. The challenger $\mathcal{C}$ randomly selects i from $\{0,1\}$, b^* from $\{0,1\}^l$, and e^* and s^* from Z_q and computes $c^* = b^* \oplus m_i$ and $w^* = (s^*)^{-1}$. When H_1 is queried at $K^* = (e^*w^* + w^*SK_S)PK_R$, the value b^* is returned directly. When H_2 is queried at $(m_i, K^* = (e^*w^* + w^*SK_S)PK_R, PK_S, PK_R)$, the value e^* is returned directly. The challenger $\mathcal{C}$ returns challenging signcryption $\sigma^* = (c^*, e^*, s^*)$ to $\mathcal{O}$. The adversary $\mathcal{O}$ initiates the second round of query, which is same as the first round of query, but the adversary $\mathcal{O}$ cannot send unsigncryption query for the signcryption result σ^*. At the end of the simulation, the adversary $\mathcal{O}$ outputs i' as the guess for i. If $i' = i$, the

challenger $\mathscr{C}$ outputs $k = ew + wSK_S$ as an answer to the ECDLP, else the challenger $\mathscr{C}$ fails to solve the ECDLP.

In the view of the adversary $\mathcal{O}$, the challenger $\mathscr{C}$ provides a simulation environment similar to the actual environment. However, in the challenge phase, the answer of H_2 to the query $(m_i, K^* = (e^*w^* + w^*SK_S)PK_R, PK_S, PK_R)$ is different. This is because m_i can only be determined at the end of the challenge phase. At this point, $q_{H_2} + q_{Sig} + q_{Uns}$ is the maximum number that H_2 is queried. Therefore, the challenger $\mathscr{C}$ has an advantage of at least $\varepsilon' \geq \varepsilon/(q_{H_2} + q_{Sig} + q_{Uns})$ to solve the ECDLP problem.

3.3.2. Unforgeability. Unforgeability is a required property of signature. Since signcryption needs to realize both signature and encryption, the signcryption scheme must also have unforgeability. According to Theorem 6, our signcryption scheme has unforgeability.

Theorem 6. *In the random oracle model, if there is an adversary $\mathcal{O}$ who can win the game of Definition 4 with the advantage of ε, there is a challenger $\mathscr{C}$ who can solve the ECDLP problem with the advantage of $\varepsilon/(q_{H_1} + q_{H_2} + q_{Sig} + q_{Uns})$. q_{H_1}, q_{H_2}, and q_{Sig} represent the number of times the adversary initiates H_1 query, H_2 query, and signcryption query, respectively.*

Proof. At the beginning of the game, the challenger $\mathscr{C}$ runs algorithm Keygen(cp, r) to generate a sender key pair (SK_S, PK_S) and a receiver key pair (SK_R, PK_R) and sends (PK_S, PK_R) to adversary $\mathcal{O}$. The challenger $\mathscr{C}$ manages three lists $L_{H_1}, L_{H_2}, L_{Sig}$, which are initially empty. L_{H_1}, L_{H_2} are used to track the adversary's queries to oracle H_1, H_2, respectively, L_{Sig} is used to simulate signcryption oracle. □

Suppose the public key of the receiver is PK_R, the adversary uses the oracle described in the proof of Theorem 5 to send various queries. After these queries, in the forgery phase, the adversary outputs the forged signcryption result. It can be seen from the proof of Theorem 5 that our simulation is equivalent to the actual attack environment. In order to forge successfully, the adversary must send H_1 query and H_2 query to get $\sigma^* = (c^*, e^*, s^*)$ corresponding to message m^*. The probability that the adversary chooses the correct record in the list L_{H_1}, L_{H_2} is $1/(q_{H_1} + q_{H_2} + q_{Sig} + q_{Uns})$, so the challenger $\mathscr{C}$ has the advantage of $\varepsilon/(q_{H_1} + q_{H_2} + q_{Sig} + q_{Uns})$ to solve ECDLP problem.

3.3.3. Integrity. Integrity means that information cannot be accidentally or maliciously deleted, modified, forged, replayed, and inserted during transmission and storage.

Theorem 7. *Our signcryption scheme has integrity.*

Proof. In our signcryption scheme, it is very difficult for an attacker to tamper with the information between the sender and receiver. Because this tampering requires the hash value b, and b corresponds to the hash value of a random point of the elliptic curve, due to the collision resistance of the hash function, the attacker cannot determine the point of the elliptic curve corresponding to the hash value b. Furthermore, every part of ciphertext $c = b \oplus m$ depends on all message blocks. Once a malicious attacker makes any change to a particular block of information, it will cause the ciphertext to change. Therefore, our signcryption scheme has integrity. □

3.3.4. Nonrepudiation. Nonrepudiation in signcryption and signature is the same. Nonrepudiation is preventing a communicating party from denying a previous promise or behavior. In a signcryption scheme, nonrepudiation means that a signer cannot deny that he signed a valid message after signing it.

Theorem 8. *Our signcryption scheme has nonrepudiation.*

Proof. In our signcryption scheme, when the sender signs message m, it first calculates the hash value of message m using its own public key PK_S and receiver's public key PK_R and then signs this hash value with his own private key SK_S. Therefore, the sender cannot deny its signature to message m. In addition, in unsigncryption, the receiver will use the sender's public key PK_S and its own public key PK_R to calculate the hash value. If it is equal to the received hash value, it means that the received signature is indeed signed by the sender. Therefore, our scheme has nonrepudiation. □

3.3.5. Availability. Availability refers to the property that all resources can be accessed by authorized parties at the appropriate time; i.e., information can be accessed by authorized entities and used on demand.

Theorem 9. *Our signcryption scheme has availability.*

Proof. In our signcryption scheme, the recipient, as an authorized entity, can use its own private key to obtain the plaintext m signed by the sender through the unsigncryption after obtaining the signcryption and then use the plaintext m to perform other required operations. Therefore, our signcryption scheme has availability. □

3.3.6. Forward Secrecy. Forward secrecy means that exposure of private key of the encryptor does not affect the confidentiality of previously encrypted messages.

Theorem 10. *Our signcryption scheme has forward secrecy.*

Proof. In our signcryption scheme, if the sender's private key is leaked, the adversary must know the value of b in order to obtain the previous session content, so he must obtain the value k. However, k is randomly selected by the sender. Even if the adversary obtains the sender's private key, he still cannot recover the plaintext information. Therefore, our signcryption scheme has forward secrecy. □

3.3.7. Internal Security. The security model of signcryption can be divided into external security and internal security. External security means that the adversary only knows

public information. Internal security means that the adversary knows the sender's or receiver's private key in addition to the public information. That is, if the sender's private key is exposed, the adversary still cannot recover the plaintext from the ciphertext; if the receiver's private key is exposed, the adversary still cannot forge the ciphertext. Obviously, internal security is stronger than external security.

Theorem 11. *Our signcryption scheme has internal security.*

Proof. On the one hand, in our signcryption scheme, if the adversary wants to recover the plaintext m from the ciphertext c, it must obtain the hash value b. Similar to Theorem 7, due to the collision resistance of the hash function and the randomness of the random number k, the adversary cannot determine the point on the elliptic curve corresponding to the hash value b. Therefore, even if the adversary possesses the sender's private key, the plaintext still cannot be recovered from the ciphertext. On the other hand, in our signcryption scheme, if the adversary possesses the receiver's private key, it is also impossible to forge the valid ciphertext c' of the plaintext m'. The reason is that even if the adversary uses SK_R to compute the value of X, gets the hash value b', and then uses $c = b \oplus m$ to get the ciphertext c' of the plaintext m', the ciphertext c' is invalid. Because the ciphertext c in the signcryption result is the encryption of the plaintext m', and the s in the signcryption result is the signature of the plaintext m, which will make the unsigncryption fail, therefore, our signcryption scheme has internal security. □

We compare the security of our signcryption scheme with Tsai's ECC-based signcryption scheme [31] and Zhou's signcryption scheme [30]. It can be seen from Table 1 that our scheme satisfies the confidentiality, unforgeability, integrity, nonrepudiation, and availability of the other two schemes and also satisfies forward secrecy and internal security. Therefore, compared with the existing signcryption schemes, our signcryption scheme is more secure.

3.4. Performance Evaluation. In this section, we compare the computational and communication overhead of our signcryption scheme with Tsai's scheme [31] and Zhou's scheme [30] in detail. Among them, the computational overhead mainly compares the calculation amount of the signcryption and unsigncryption algorithms, and the calculation amount mainly counts the execution times of the point multiplication operation, point addition operation, number multiplication operation, and inversion operation. The XOR operation, Hash operation, and the number addition operation are not counted. The computational overhead and communication overhead of the three schemes are shown in Table 2. In this table, PM, PA, NM, and IN represent the point multiplication operation, point addition operation, number multiplication operation, and inversion operation, respectively; l_m represents the length of the plaintext message; $|G|$ represents the length of the element on the group; and $|Z_n^*|$ represents the length of the element in Z_n^*.

Among the various operations counted in Table 2, the point multiplication operation takes the most time, followed by the point addition operation. It can be seen from the calculation amount in Table 2 that the computational overhead of our scheme is much less than that of the other two schemes. In addition, the communication overhead of our scheme is comparable to Zhou's scheme and smaller than Tsai's scheme. Our scheme has forward security and internal security in addition to the same confidentiality, unforgeability, integrity, nonrepudiation, and availability as the other two schemes. Overall, our scheme is an efficient and secure ECC-based signcryption scheme.

4. Our Key Management Scheme

In this section, our ECC-based signcryption scheme will be applied to the key management scheme in the smart lock system. In Subsection 4.1, we recall the model of the smart lock system. In Subsection 4.2, we give an overview of the key management scheme for the smart lock system. In Subsection 4.3, we use the above ECC-based signcryption scheme as a building block to construct our key management scheme of the smart lock system. Finally, in Subsection 4.4, we observe the bit-oriented overhead of our smart lock key management scheme through experimental simulations.

4.1. The Model of the Smart Lock System. There are three main parties in a smart lock system [40], that is, smart phone (SP) of user, smart lock (SL), and management server (MS), which is shown in Figure 1. Among them, MS receives the request from the user's SP, reviews the user's qualification, receives the operation information of SL, manages the unlock key, and helps SL and SP exchange the public key. SL communicates with MS through narrow band Internet of Things (NB-IOT) and receives the signcryption for the unlock key. SP of the legitimate user applies for the unlock key to MS and sends the signcryption of this unlock key to SL through Bluetooth.

In the smart lock system, MS is trusted, which cannot disclose the unlock key to the adversary. MS will send correct unlock key to SL and legitimate user and cancel the unlock key of the expired user. SL is safe, controllable, and will not disclose the unlock key. The user is semi-honest. Although he will follow the rule of the key management scheme, he will try to use the obtained information to unlock when his key expires or he has no key.

Each smart lock has a unique international mobile equipment identity (IMEI), which is a 15-digit "electronic serial number." In this paper, the IMEI will be used to generate a session key for the smart lock.

4.2. The Overview of Key Management Scheme. In the key management scheme of the smart lock system, MS and SL generate their own private keys, respectively, and then calculate their own public keys through ECC and send the public key to each other. They realize the key exchange and generate the shared session key between them. MS generates the unlock key, uses the session key to encrypt the unlock key, and sends the encryption result to SL. SL uses the session

Table 1: Security comparison of three signcryption schemes.

	Confidentiality	Unforgeability	Integrity	Nonrepudiation	Availability	Forward secrecy	Internal security
Tsai's scheme	Y	Y	Y	Y	Y	N	N
Zhou's scheme	Y	Y	Y	Y	Y	N	N
Our scheme	Y	Y	Y	Y	Y	Y	Y

Note: "Y" means that the scheme has this property; "N" means that the scheme does not have this property.

Table 2: Performance comparison of three signcryption schemes.

	Computational overhead		Communication overhead
	Signcryption	Unsigncryption	
Tsai's scheme	$4PM + 2NM$	$3PM + 3PA$	$l_m + 3\|G\| + \|Z_n^*\|$
Zhou's scheme	$3PM + 2PA + 2NM + 1\text{IN}$	$6PM + 5PA$	$l_m + 2\|Z_n^*\|$
Our scheme	$1PM + 1NM + 1\text{IN}$	$2PM + 1PA + 1\text{IN} + 2NM$	$l_m + 2\|Z_n^*\|$

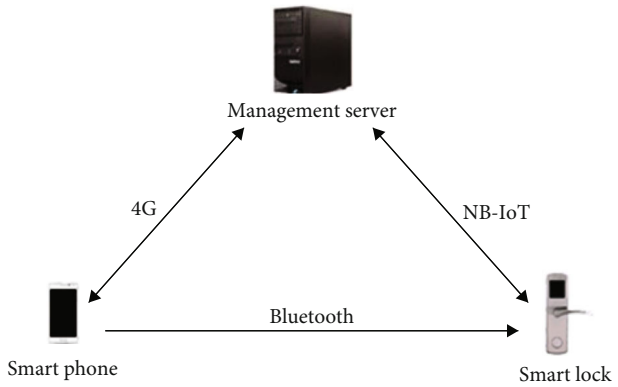

Figure 1: Architecture of smart lock system.

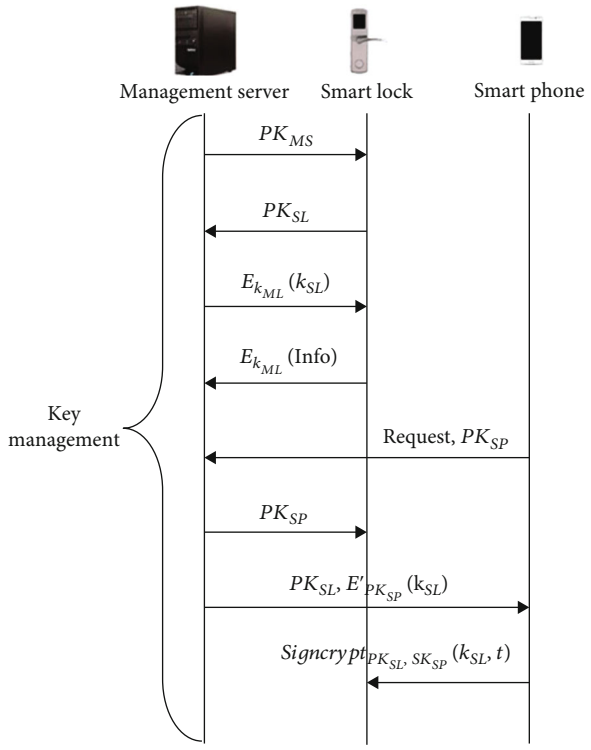

Figure 2: Flow chart for key management of smart lock system.

key to encrypt its operation information and uploads the encryption result to MS. This two communications adopt AES symmetric encryption through the Nb-IOT. After the user applies to MS for the house, MS reviews the user's qualification. If the user does not meet the conditions, MS refuses to send the key to him. If the user meets the conditions, MS sends the user's public key to SL. At the same time, MS uses the user's public key to encrypt the unlock key and sends the encryption result and the public key of SL to the user. The user uses his private key to decrypt the unlock key. This communication uses elliptic curve public key cryptosystem. After that, the user can use the received public key of SL and his own private key to signcrypt the unlock key and send the signcryption result to SL. SL uses the received public key of user and his own private key to de-signcrypt the signcryption, thus obtaining the unlock key. The process above uses our ECC-based signcryption scheme. Finally, SL compares the unlock key from the user with its own unlock key. If the two unlock keys are different, SL cannot be unlocked.

During the lifetime, the smart lock system can periodically update the key according to the security status. If the user loses the housing qualification, MS regenerates the unlock key and sends it to SL. The user cannot unlock with the old key.

4.3. Our Key Management Scheme. Our key management scheme is detailed as follows:

(1) Key exchange between MS and SL. MS selects private key SK_{MS}. SK_{MS} is confidential and satisfying $SK_{MS} < n$. MS computes public key $PK_{MS} = SK_{MS} \times P$ and sends PK_{MS} to SL through NB-IOT. In the transmission, even if PK_{MS} is attacked, the adversary

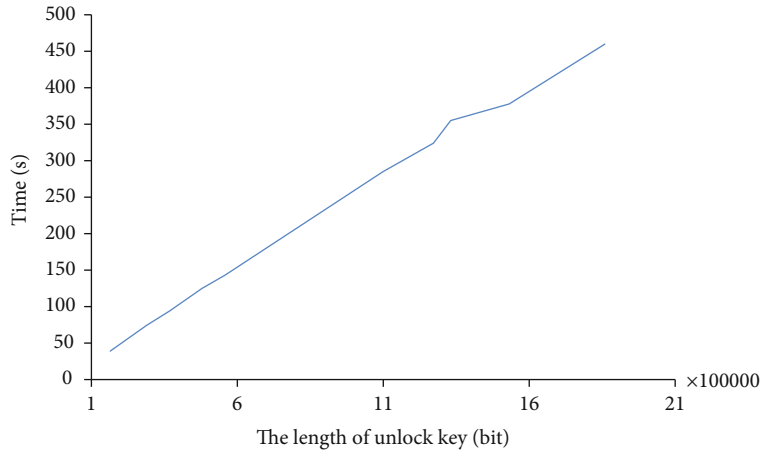

Figure 3: The performance of our smart lock key management scheme.

cannot calculate SK_{MS} by the known P since ECDLP problem

(2) SL selects private key SK_{SL}. SK_{SL} is confidential and satisfying $SK_{SL} < n$. SL computes public key $PK_{SL} = SK_{SL} \times P$ and sends PK_{SL} to MS through NB-IOT. In the transmission, even if PK_{SL} is attacked, the adversary cannot calculate SK_{SL} by the known P since ECDLP problem

(3) After the MS receives PK_{SL}, it uses the private key SK_{MS} and the received PK_{SL} to generate the secret key

$$K = SK_{MS} \times PK_{SL} = SK_{MS} \times SK_{SL} \times P = (x_K, y_K). \quad (3)$$

Similarly, SL uses the private key SK_{SL} and the received PK_{MS} to generate the secret key

$$K = SK_{SL} \times PK_{MS} = SK_{SL} \times SK_{MS} \times P = (x_K, y_K). \quad (4)$$

The two secret keys K are equal, which only are known as MS and SL. Because the secret key K is a pair of numbers (x_K, y_K), MS and SL can select the session key $k_{ML} = x_K + last(IMEI)$ according to the factory agreement, and function $last(IMEI)$ is the last digit of IMEI of SL. Because x_K is known only by MS and SL, the session key k_{ML} is also known only by them.

(4) MS generates 128 *bit* unlock key $k_{SL} = random()$ and sends encryption result $E_{k_{ML}}(k_{SL})$ to SL. At the same time, SL sends the encryption result $E_{k_{ML}}(\text{Info})$ to MS and reports the operation information $Info$, where E is AES symmetric encryption algorithm and $random()$ is random generating function

(5) Users download APP through their SP. SP selects private key SK_{SP}. SK_{SP} is confidential and satisfying $SK_{SP} < n$. SP computes public key $PK_{SP} = SK_{SP} \times P$, sends public key PK_{SP} and the request for unlock key to MS. MS will review the user's qualification after receiving the user's request. If the user does not have the housing qualification, the MS rejects his request

(6) If the user has the housing qualification, MS encrypts the unlock key with the received public key of the user and gets the ciphertext

$$C = E'_{PK_{SP}}(k_{SL}) \quad (5)$$

where E' is elliptic curve public key cryptosystem. Then, MS sends this ciphertext and the public key PK_{SL} to SP and sends the public key PK_{SP} to SL at the same time. In the transmission, even if C is overheard by the adversary, the adversary cannot get k_{SL} by decrypting C since the private key SK_{SP} is not known.

(7) After receiving the ciphertext C and the public key PK_{SL}, the SP uses its private key SK_{SP} to calculate

$$D_{SK_{SP}}(\text{C}) = D_{SK_{SP}}\left[E_{PK_{SP}}(\text{k}_{SL})\right] = \text{k}_{SL}. \quad (6)$$

Thus, the SP obtains the unlock key k_{SL} of SL.

(8) SP uses our ECC-based signcryption algorithm to generate $Signcrypt_{PK_{SL},SK_{SP}}(k_{SL}, \text{t})$ of unlock key k_{SL} and time stamp t, sends the signcryption to SL through Bluetooth. SL uses PK_{SP} and SK_{SL} to calculate

$$DeSigncrypt_{\text{KU}_{SP},SK_{SL}}\left[Signcrypt_{PK_{SL},SK_{SP}}(k_{SL}, t)\right] = (k_{SL}, \text{t}). \quad (7)$$

After getting the unlock key$k_{SL}$and time stamp$t$, SL checks them. If the unlock key is wrong, SL will not unlock. Our ECC-based signcryption scheme plays the role of encryption and authentication at the same time. The addition of time stamp can prevent replay attack.

(9) If the user loses the housing qualification, MS generates a new unlock key and sends it to SL. As a result, the SP cannot unlock with its old key k_{SL}

The flow chart of our key management is shown in Figure 2.

4.4. Performance Analysis. In this subsection, we observe the bit-oriented overhead of our smart lock key management scheme through experimental simulations. Here, AES symmetric encryption is performed in ECB mode, and the q in the elliptic curve used in our signcryption scheme and the order q of the group G_1 are both 160 bit. The experimental environment is as follows: AMD Ryzen 7 5800H, reference frequency 3.20 GHz, memory 16GB (DDR4-3200 MHz), and Windows 11 operating system.

As can be seen from Figure 3, when the length of the unlock key is as high as$21 \times 10^5 bit$, the time consumed of our key management scheme does not exceed$500\,s$. It is worth noting that in the actual deployment of the smart lock key management system, the length of the unlock key is generally not so long. Therefore, our key management scheme is practical and efficient.

5. Conclusion

In this paper, we designed an efficient and secure ECC-based signcryption scheme. Our signcryption scheme has been highly efficient and satisfies multiple security properties; it can also be used for mobile device authentication. Unfortunately, our signcryption scheme is only suitable for single-factor authentication. In the future, it will be interesting to consider applying our signcryption scheme to other application scenarios.

In addition, we proposed a practical and efficient key management scheme of the smart lock using our signcryption scheme firstly. Our key management scheme does not require manually memorizing the unlocking key, and every time the unlocking key is different, not being fixed. However, there has been a recent trend to study smart lock systems using deep learning methods. In addition to hand gesture recognition, face recognition is gradually popular. In the future, we will consider how to use deep learning methods in smart lock key management systems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was sponsored by the National Natural Science Foundation of China (No.11401172), the Science and Technology Project of Henan Educational Committee of China (No.20A520012).

References

[1] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) << cost(signature) + cost(encryption)," in *Advances in Cryptology – CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pp. 165–179, 1997.

[2] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[3] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology-CRYPTO'85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, pp. 417–426, 1985.

[4] Y. Zheng and H. Imai, "How to construct efficient signcryption schemes on elliptic curves," *Information Processing Letters*, vol. 68, no. 5, pp. 227–233, 1998.

[5] R. Steinfeld and Y. Zheng, "A signcryption scheme based on integer factorization," in *Information Security, Third International Workshop, ISW 2000, Proceedings*, pp. 308–322, 2000.

[6] B. Libert and J.-J. Quisquater, "A new identity based signcryption scheme from pairings," in *Proceedings 2003 IEEE Information Theory Workshop, ITW 2003, La Sorbonne, Paris, France, 31 March -4 April, 2003*, pp. 155–158, 2003.

[7] J. Malone-Lee, "Identity-based signcryption," *IACR Cryptology ePrint Archive*, vol. 2002, p. 98, 2002.

[8] X. Boyen, "Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography)," in *Advances in Cryptology-CRYPTO 2003, Proceedings*, pp. 383–399, 2003.

[9] J. K. Liu, J. Baek, and J. Zhou, "Online/offline identity-based signcryption revisited," in *Information Security and Cryptology -6th International Conference, Inscrypt 2010, Shanghai, China, October 20-24, 2010, Revised Selected Papers*, pp. 36–51, 2010.

[10] S. S. D. Selvi, S. S. Vivek, and C. P. Rangan, "Identity based public verifiable signcryption scheme," in *Provable Security -4th International Conference, ProvSec 2010. Proceedings*, pp. 244–260, 2010.

[11] S. S. D. Selvi, S. S. Vivek, D. Vinayagamurthy, and C. P. Rangan, "ID based signcryption scheme in standard model," in *Provable Security -6th International Conference, ProvSec 2012. Proceedings*, pp. 35–52, 2012.

[12] A. W. Dent, "Hybrid signcryption schemes with insider security," in *Information Security and Privacy, 10th Australasian Conference, ACISP2005, Brisbane, Australia, July 4-6, 2005, Proceedings*, pp. 253–266, 2005.

[13] T. E. Bjørstad and A. W. Dent, "Building better signcryption schemes with tag-kems," in *Public Key Cryptography-PKC 2006, Proceedings*, pp. 491–507, 2006.

[14] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2008, Tokyo, Japan, March 18-20, 2008*, pp. 369–372, 2008.

[15] T. Pandit, S. K. Pandey, and R. Barua, "Attribute-based signcryption: Signer privacy, strong unforgeability and IND-CCA2 security in adaptivepredicates attack," in *Provable Security -8th International Conference, ProvSec 2014. Proceedings*, pp. 274–290, 2014.

[16] P. Datta, R. Dutta, and S. Mukhopadhyay, "Compact attribute-based encryption and signcryption for general circuits from multilinear maps," in *Progress in Cryptology-INDOCRYPT 2015-16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings*, pp. 3–24, 2015.

[17] P. Datta, R. Dutta, and S. Mukhopadhyay, "Functional signcryption: notion, construction, and applications," in *Provable Security -9th International Conference, ProvSec 2015, Proceedings*, pp. 268–288, 2015.

[18] W. Yang, M. Manulis, A. Man Ho, and W. Susilo, "Relations among privacy notions for signcryption and key invisible "sign-then-encrypt"," in *Information Security and Privacy -18th Australasian Conference, ACISP2013, Brisbane, Australia, July 1-3, 2013. Proceedings*, pp. 187–202, 2013.

[19] D. Nalla and K. C. Reddy, "Signcryption scheme for identity-based cryptosystems," *IACR Cryptology ePrint Archive*, vol. 2003, p. 66, 2003.

[20] B. Libert and J.-J. Quisquater, "Improved signcryption from q-diffiehellman problems," in *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, pp. 220–234, 2004.

[21] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology-EUROCRYPT 2005, Proceedings*, pp. 114–127, 2005.

[22] Y. Yong, B. Yang, Y. Sun, and S. Zhu, "Identity based signcryption scheme without random oracles," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 56–62, 2009.

[23] J. Kar, "An efficient signcryption scheme from q-diffie-hellman problems," *IACR Cryptology ePrint Archive*, vol. 2012, p. 483, 2012.

[24] B. Libert and J.-J. Quisquater, "Efficient signcryption with key privacy from gap diffie-hellman groups," in *Public Key Cryptography-PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*, pp. 187–200, 2004.

[25] J. Kar, "Provably secure identity-based aggregate signcryption scheme in random oracles," *IACR Cryptology ePrint Archive*, vol. 2013, p. 37, 2013.

[26] L. I. U. Zhenhua, L. I. Juanjuan, and Z. U. Longhui, "Revocable id-based signcryption scheme," *Journal of Sichuan University (Engineering Science Edition)*, vol. 46, no. 2, pp. 79–86, 2014.

[27] A. Braeken and P. Porambage, "Efficient generalized signcryption based on ecc," *International Journal on Cryptography and Information Security (IJCIS)*, vol. 5, no. 2, pp. 1–13, 2015.

[28] J. Kar and S. Naik, "Generic construction of certificateless signcryption scheme," *IACR Cryptology ePrint Archive*, vol. 2016, p. 318, 2016.

[29] H. A. N. Yiliang, L. I. Chong, F. A. N. G. Dingyi, and Y. A. N. G. Xiaoyuan, "New signcryption scheme based on niederreiter cryptosystem," *Journal of Sichuan University (Engineering Science Edition)*, vol. 48, no. 2, pp. 97–103, 2016.

[30] Z. H. O. U. Yan-Wei, Y. A. N. G. Bo, and W. A. N. G. Qing-Long, "Secure certificateless signcryption scheme without bilinear pairing," *Journal of Software*, vol. 28, no. 10, pp. 2757–2768, 2017.

[31] C.-H. Tsai and S. Pin-Chang, "An ecc-based blind signcryption scheme for multiple digital documents," *Security and Communication Networks*, vol. 2017, Article ID 8981606, 14 pages, 2017.

[32] H. Dai, D. Wang, J. Chang, and X. Maozhi, "On the RCCA security of hybrid signcryption for internet of things," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8646973, 11 pages, 2018.

[33] A. W. Dent, "Hybrid signcryption schemes with outsider security," in *Information Security, 8th International Conference, ISC 2005, Singapore, September 20-23, 2005, Proceedings, volume 3650 of Lecture Notes in Computer Science*, pp. 203–217, 2005.

[34] D. He, D. Wang, and W. Shuhua, "Cryptanalysis and improvement of a password-based remote user authentication scheme without smart cards," *Information technology and control*, vol. 42, no. 2, pp. 170–177, 2013.

[35] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.

[36] J. I. A. Rong-yuan, W. A. N. G. Yi-huai, and W. A. N. G. Xiaoning, "Lightweight encryption algorithm for narrowband internet of things," *Computer Engineering and Design*, vol. 39, no. 10, pp. 3039–3044, 2018.

[37] L. Chih-Chung and S.-Y. Tseng, "Integrated design of AES (advanced encryption standard) encrypter and decrypter," in *13th IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP 2002), 17-19 July 2002, San Jose, CA, USA*, pp. 277–285, IEEE Computer Society, 2002.

[38] Z. H. A. N. G. Huan-lan and X. I. A. O. Ming-bo, "Design of intelligent lock system based on 433mhz band security," *Computer Engineering and Design*, vol. 39, no. 9, pp. 2736–2742, 2018.

[39] L. I. U. Meng-jun, S. H. A. Tao, L. I. Dan, and L. I. U. Shu-bo, "Reliable security lock key updating scheme over narrow band internet of things," *Computer Science*, vol. 46, no. 4, pp. 137–143, 2019.

[40] S. Tao, L. Mengjun, L. Dan, and L. Shubo, "Nb-iot security smart lock system solution under background of public rental housing," *Application Research of Computers*, vol. 36, no. 6, pp. 1797–1802, 2019.

[41] Y. Wang, Y. Shu, M. Xiuqian Jia, L. X. Zhou, and L. Guo, "Multifeature fusion-based hand gesture sensing and recognition system," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.