RESEARCH ARTICLE

A secure and efficient certificateless content extraction signature with privacy protection

Can Zhao¹, Jiabing Liu¹, Fuyong Zheng², Dejun Wang¹, Bo Meng^{1*}¹ School of Computer Science, South-Central University for Nationalities, Wuhan, China, ² Information and communication Branch, Jiangxi electric power Co. Ltd., Nanchang, China* mengscuec@gmail.com

Abstract

Efficiency and privacy are the key aspects in content extraction signatures. In this study, we proposed a Secure and Efficient and Certificateless Content Extraction Signature with Privacy Protection (SECCESPP) in which scalar multiplication of elliptic curves is used to replace inefficient bilinear pairing of certificateless public key cryptosystem, and the sign-cryption idea is borrowed to implement privacy protection for signed messages. The correctness of the SECCESPP scheme is demonstrated by the consistency of the message and the accuracy of the equation. The security and privacy of the SECCESPP scheme are demonstrated based on the elliptic curve discrete logarithm problem in the random oracle model and are formally analyzed with the formal analysis tool ProVerif, respectively. Theory and experimental analysis show that the SECCESPP scheme is more efficient than other schemes.

OPEN ACCESS

Citation: Zhao C, Liu J, Zheng F, Wang D, Meng B (2021) A secure and efficient certificateless content extraction signature with privacy protection. PLoS ONE 16(11): e0258907. <https://doi.org/10.1371/journal.pone.0258907>

Editor: Pandi Vijayakumar, University College of Engineering Tindivanam, INDIA

Received: July 29, 2021

Accepted: October 7, 2021

Published: November 1, 2021

Copyright: © 2021 Zhao et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript.

Funding: This work was supported in part by the Fundamental Research Funds for the Central Universities, South-Central University for Nationalities No. CZZ21001, No.CZT20015 and QSZ17007; the National Key R&D Program of China No. 2020YFC1522900; and in part by the National Natural Science Foundation of China No.62062019; the natural science foundation of Hubei Province under the grants No.2019CFB815 and No.2018ADC150.

Introduction

Content extraction signatures have been widely used to protect electronic medical records and electronic commerce security, in which the signature verifier can verify the authenticity of the extracted message without knowing the entire signed message [1]. Efficiency and privacy protection are the key aspects of a content extraction signature. For efficiency purposes, most content extraction signature schemes were developed using the traditional public key cryptosystem [2–5] or the identity-based public key cryptosystem [6–10]. Owing to the problems of certificate management in traditional public key cryptosystems and user key management in identity-based public key cryptosystems, respectively, people have turned to certificateless public key cryptosystems to implement the content extraction signatures [11–13] in the present day because no certificate is required and key management problems barely exist for certificateless public key cryptosystem. However, certificateless public key cryptosystem are based on costly bilinear pairing of elliptic curves. The efficiency of such an approach is low. The operations currently available for elliptic curves include exponential operation, scalar multiplication, bilinear pairing and a hash function According to references [8,9], scalar multiplication is the most efficient operation and is therefore a good option to replace inefficient bilinear pairing in certificateless public key cryptosystem.

Competing interests: The authors have declared that no competing interests exist.

For privacy protection, the signed message of a content extraction signature may contain a private message. However, existing content extraction signatures do not consider privacy protection. The available privacy protection methods mainly include data distortion and data encryption [11]. Data distortion has no universality and depends on the data types used. Data encryption is a good choice for implementing privacy with regard to content extraction signatures. In general, the sign-then-encrypt approach can provide privacy and authentication. However, it is inefficient because the encryption and signature processes are separate operations. But signcryption [2] can provide privacy and authentication with high efficiency because the encryption and signature are completed in a single logical step.

Therefore, to improve the efficiency of and provide privacy protection for content extraction signatures, we propose a Secure and Efficient Certificateless Content Extraction Signature with Privacy Protection (SECCESPP) scheme. The main works is presented as follows:

1. Scalar multiplication is used on elliptic curves to replace inefficient bilinear pairing in a certificateless public key cryptosystem, and the signcryption idea from data encryption approach is borrowed to implement privacy for signed message.
2. The correctness of the scheme is demonstrated by the consistency of the message and the accuracy of the equation, the scheme's security is verified based on the elliptic curve discrete logarithm problem in the random oracle model, and privacy is analysed with the formal analysis tool ProVerif.
3. The scheme is compared to the related schemes [14–19], revealing that the SECCESPP scheme is more efficient.

The rest of the paper is organized as follows. Section 1 discussed the related works of privacy protection, content extraction signature and signcryption. Section 2 makes a simple review of preliminaries. Section 3 presents the SECCESPP scheme. Section 4 illustrates the correctness of the SECCESPP scheme. Section 5 gives proofs about security requirements. Section 6 makes comparisons of performance. Finally, Section 7 presents the conclusion and future work.

Related work

Since the SECCESPP scheme involves privacy protection, content extraction signatures and signcryption, this section briefly introduces the related works regarding these three components.

Privacy protection

There is much information that is vulnerable to attackers. In 2017, Adat et al. discussed the history, background, statistics of IoT and security based analysis of IoT architecture and provided taxonomy of various defense mechanisms [20]. It is urgent to protect private information. The existing privacy protection technologies can be divided into two categories: data distortion and data encryption approaches [11].

Technologies based on data distortion distort sensitive data while keeping some data or data attributes unchanged, and the most common technique of this type is the differential privacy technique. In 2018, Ye [21] proposed a new localized differential privacy protection model. The protection model adopts random response technology, which first makes the data private and then sends it, to provide comprehensive protection for sensitive information; this approach can not only resist attackers with arbitrary background knowledge but also prevent privacy attacks from distrusted third parties. However, it depends heavily on the

accompanying data, and different algorithms are designed for different data, so it has no universality. In 2021, Stergoiu et al. [22] proposed an innovative system of secure caching scenario which operates in a wireless-mobile 6G network for managing BD on Smart Buildings(SB) and created a novel and secure Cache Decision System(CDS) in a wireless network that operates over a SB, which offer the users a safer and efficient environment for browsing the internet, sharing and managing large-scale data in the fog. It could be a start point for better and more efficient wireless networking scenario, for managing and sharing Big Data on a Smart Building.

Technologies based on data encryption hide sensitive data in the process of data mining. It is mainly used in distributed application environments and can solve the communication security problem. At present, Homomorphic Encryption (HE) is widely used. HE refers to specific classes of encryption schemes that allow for computing directly on encrypted data without having to decrypt them. In 2020, Zhao [23] presented a circular secure public key homomorphic encryption scheme using noise flooding technique, and provided security proofs and parameter setting. Furthermore, by introducing the refuse sampling technique, an optimized circular secure public key homomorphic encryption scheme was given, and the system parameters were reduced from the super-polynomial level to the polynomial level, thereby greatly reducing the size of the public key and ciphertext. Then, the computational complexity of ciphertext evaluation could be effectively improved, and the performance of the homomorphic encryption scheme could be enhanced. However, at the same time, there are some problems with this method, such as high computing costs, high communication costs, complex deployment, and high practical application difficulty. In 2021, Zhang et al. [24] proposed a secure decentralized spatial crowdsourcing scheme for 6G-Enabled Network in Box using CBC-MAC authenticated encryption mechanism to provide confidentiality and integrity. It solves leakage of sensing nodes locations. But it Still hasn't solved the problem of data leaks in transit. In the same year, in order to solve the security of shared information VANET system, Vijayakumar et al. [25] proposed an efficient batch authentication and key exchange schemes, which will be applied to blockchain users in the future. Then, Azees et al. [26] completed the following work, applying blockchain technology to the security guarantee of VANET system, realizing the rapid reauthentication of vehicles, and making a contribution to the information security in the future blockchain era.

Content extraction signature

The Content Extraction Signature (CES) was first proposed by Steinfeld [27]. According to the technology used, CES types are mainly divided into a CommitVector (CV)-based content extraction signature, an RSA-based content extraction signature, and a hash tree (HT)-based content extraction signature.

A CES scheme [3] based on a CV has the characteristics of unforgeability and exclusivity. Its unforgeability is jointly guaranteed by the standard digital signature EUF-CMA and the binding of the message commitment scheme, while its exclusivity is guaranteed by the hiding of the message commitment scheme. Scheme [27] formalized and proved these two securities. Scheme [3] has a lower computational cost than the CES scheme [27], which requires a signature operation and a commitment operation to be performed for the original signature generation process. However, because the original signature and the intercepted signature contain the committed random numbers of all the retained sub-data and the committed values of the deleted sub-data, the length of the signature expands, and the communication overhead increases.

To solve the problem of CVs, an RSA-based CES was proposed. An RSA-based CES is formed on the basis of a CV-based CES using the RSA signature, and the length of the signature is only the length of the modulo of the RSA signature, which greatly reduces the length of the signature. Combining this with the idea of batch signatures, in 2014, Li [4] proposed an improved scheme for content extraction signatures based on RSA. The scheme can judge whether a content extraction access structure (CEAS) meets the given extraction conditions through the correspondence between (M') and $CI(M')$. In 2015, Lan [28] proposed an identity-based CES scheme. This scheme does not need to sign every sub-message, thereby improving efficiency, and it can prevent PKG from forging signatures and thus improve its application value. In 2017, Wang [29], based on [28], achieved the goal of shortening the length of the signature by reducing the commitment value and random number and performed unified signature and verification operations on sub-messages, which improved the efficiency of signing and verification.

However, by using quantum cryptography, the keys of Std.RSA might get broken down to approximately 850 bits. This result in the need to enhance the current public key cryptosystem. Thus, the HT-based CES was proposed. Drawing on the idea of a binary tree, hashing every two message blocks generates a commitment hash value, recursively, layer by layer, and obtains a total hash value, which greatly reduces the chance of the CES breaking. In 2016, Thirumalai [30] proposed a commitment tree-based batch signature scheme. Compared with the CV-based CES and RSA-based CES, this scheme has a lower signature length, fewer calculation operations, and improved signature efficiency. In 2019, Szalachowski [31] proposed a TLS-N method based on the TLS extension. In this method, the Merkle tree is used during the process of generating evidence. The server generates evidence about the TLS session content, generates a noninteractive certificate about the TLS session content on the client, and then sends the session content and the certificate to a third party for verification. In 2020, Cheng [32] proposed a blockchain based secure storage and sharing scheme for electronic health records data. In this scheme, a certificateless content extraction signature algorithm is used to provide privacy protection, secure sharing of data has realized in combination with smart contracts. The combination of blockchain and content extraction signature is better applied in the electronic medical records.

Signcryption

Signcryption is a cryptographic primitive that captures a common practical scenario where one simultaneously requires confidentiality and nonrepudiation for transmitted data. Signcryption schemes achieve confidentiality and authentication simultaneously by combining public key encryption and digital signatures, offering better overall performance and security than other schemes [11]. There are three types of signcryption: public key infrastructure (PKI)-based signcryption, identity (ID)-based signcryption, and certificateless signcryption.

A PKI is required to manage and distribute public keys. In such systems, a public key is bound to the corresponding unique user ID. Trusted third-party tools are used to bind the users to unique public keys through an appropriate registration process. Based on the PKI concept, in 1997, Zheng [2] proposed signcryption, which has since been widely discussed and studied. The original scheme used interactive zero-knowledge proof technology, which is not efficient. In 2019, Yan [5] proposed a signcryption scheme that directly uses the sender's public key to verify the validity of the signature. Compared with the sign-then-encrypt mechanism, the public key size and the computational cost of the signcryption operation are both obviously reduced. However, the use of an additional third-party application makes public key cryptography expensive and inefficient.

To overcome the problem of the PKI management system, many ID-based signcryption schemes have been proposed. The idea of ID-based signcryption was first proposed by Malone [6] in 2002 along with a security model. This model was developed by Boyen [7]. Three new security notions were added: ciphertext unlinkability, ciphertext authentication and ciphertext anonymity. In 2019, Wang [8] proposed a basic model for ID-based signcryption schemes that can use bilinear pairs to design signcryption schemes. In the same year, Shankar [8] pointed out that scheme [8] was not secure and proposed three new secure solutions. However, they do not satisfy public verifiability and forward security at the same time. In response to this problem, in 2019, Pan [10] proposed a solution that uses two private keys for signcryption and unsigncryption. In 2019, Deng [33] proposed a new ID-based signcryption model, that solved the problem in which [9] does not simultaneously satisfy public verifiability and forward security. However, the system needs a third-party application for private key management to generate them secretly and distribute them to users.

To address PKI-based and ID-based signcryption issues, a certificateless signcryption approach was proposed by Riyamin [11] in 2008. It presents stronger security properties than one might expect from its internal building blocks; sharing randomness between encryption and signature modules not only provides extra savings in terms of the computational and bandwidth loads but also yields strong insider security guarantees. In 2019, Gao [12] proposed an improved certificateless signcryption scheme. This scheme guarantees the security of the signcryption phase by defining the length of the message space during the system establishment phase. In the same year, Wang [13] proposed the definition of a blind signcryption scheme under certificated and certificateless public key cryptosystems and proposed a blind signcryption scheme based on bilinear pairing. The scheme increases blindness, but the computational cost does not increase significantly. In 2020, Fang [34] proposed a certificateless multi-receiver multi-message simultaneous broadcast signcryption scheme. Combined with random elements in an elliptic curve cyclic group, the encryption key is generated, which solves the problems of receiver decryption ciphertext and identity anonymity protection. However, the scheme lacks a security mechanism when verifying the signcryption.

Preliminaries

This section introduces the commitment scheme, salt tree, binary commitment tree and content extraction access structure used in the SECCESPP scheme.

Commitment

Commitment, with two characteristics of Hiding and Binding, is a fundamental model in the field of cryptography. Hiding means that the commitment can hide information, that is, no other entity can obtain information from the commitment except the entity that places the information there. Binding means that no entity is allowed to change the information within the commitment, and it can verify that the information it receives is indeed the information originally promised.

The commitment scheme is composed of three algorithms: *Gen()*, *Com()* and *Ver()*.

Initialization phase: *Gen()* accepts a “1” bit string of length k as input and outputs a common reference string *crs*.

$$crs \leftarrow Gen(1^k) \quad (1)$$

Commitment phase: *Com()* accepts a common reference string *crs* and a committed

message m as input, and outputs m 's commitment value com and decommitment information dec .

$$(com, dec) \leftarrow Com(crs, m) \quad (2)$$

Decommitment phase: The sender sends dec and m to the receiver.

Verification phase: $Ver()$ accepts the message m , common string crs , commitment value com and decommitment information dec as input, and it outputs verification result.

$$Yes/No \leftarrow Ver(crs, com, dec, m) \quad (3)$$

In the SECCESPP scheme, the entire message M is divided into n blocks, $M = \{M[1], M[2], \dots, M[i], \dots, M[n]\}$, along with a random number string called salt, and used as input for the message commitment algorithm $C()$. Because the entropy of the message block is low, pseudorandom salt can be used to protect the committed privacy and avoid brute force attacks. Combine the message block $M[i]$ and pseudorandom salt S_{i,r,i_c} together to generate the commitment $c = C(M[i], S_{i,r,i_c})$. Recalculate the commitment $c' = C(M'[i], S_{i,r,i_c})$ for the new message $M'[i]$ during the verification phase. If $c = c'$ is true, the new message $M'[i]$ is indeed the message $M[i]$ originally promised.

Salt tree

The leaf nodes in a salt tree are used as salt to generate a binary commitment tree to greatly increase the computations required by an attack.

To protect privacy, S_{i,r,i_c} and S'_{i,r,i'_c} must remain independent, meaning that the salt tree generation process cannot be sent directly to the verifier and S_{i,r,i_c} must be retrieved. The salt tree is constructed by the function $E()$, and the salt value is obtained as the input of the binary commitment tree. The inputs for the function $E()$ are a session-based secret value and a Nonce random value, and the output is a pseudorandom salt. During the verification process, the corresponding salt is required for completion.

Binary commitment tree

The purpose of building the binary commitment tree is to include the commitment in the proof. The commitment value generated by the commitment algorithm serves as input for obtaining the leaf node value of the binary commitment tree. The hash value h_i of the binary commitment tree is calculated by the anti-collision hash function of the hash value, session message length l_r and signer information of O_i its child node. O_i is the i -th member of the order vector. The commitment values generated in section II.B are used as leaf nodes to generate the hash chain. All the leaf nodes can be verified by signature verification of the root node using the binary commitment tree.

The generation process for a binary commitment tree is as follows: A session message Record i consists of n sub-message blocks. Each message block and its corresponding salt leaf node $salt$ secret serve as input; the commitment value c is generated as the leaf node of the binary commitment tree. Every two leaf nodes are hashed in a cascade recursively, layer by layer, until the root node is finally obtained. At this point, the binary commitment tree construction process is completed.

Content extraction access structure

To extract sub-message blocks from the original message, the concept of a content extraction access structure (CEAS) is introduced. Sub-message block numbers in a CEAS must be

Fig 1. The main research framework of the SECCESP scheme.

<https://doi.org/10.1371/journal.pone.0258907.g001>

extracted after the original message is partitioned. For example, $\{M[1], M[2], M[3], M[4]\}$ means that the original message M is divided into four sub-message blocks; $CEAS = \{1, 3\}$ means that sub-message blocks $M[1]$ and $M[3]$ must be extracted, and sub-message blocks $M[2]$ and $M[4]$ can be extracted. If $CI(M') = \{1, 3, 4\}$, then $M = \{M[1], ?, M[3], M[4]\}$, where '?' represents the unextracted sub-message block, which is denoted as $CEAS \in CI(M')$. Such an extraction method is legal. If $CI(M') = \{3, 4\}$, then $M' = \{?, ?, M[3], M[4]\}$, and this is represented as $CEAS \notin CI(M')$; this kind of extraction method is illegal.

The SECCESP scheme

The SECCESP scheme can implement a content extraction signature with privacy protection for a single message in a single security mechanism in which scalar multiplication on elliptic curves and the signcryption idea are used.

Fig 1 shows the main research framework of the SECCESP scheme. The SECCESP scheme is composed of Key-Generation, Signature-Generation, Signcryption-Extraction, and Signcryption-Verification algorithms. Three roles are defined: signer, signcryptor and verifier. The signer first divides the entire message into n blocks and then generates a commitment for each block. Next, the signer generates a signature and sends it to the signcryptor. After receiving the signature, the signcryptor extracts the extracted message blocks and the corresponding commitments, encrypts the extracted message, and generates the content extraction signcryption sent to the verifier. The verifier receives and verifies the content extraction signcryption.

Key-Generation algorithm

The Key-Generation algorithm generates a public key SK_A and a private key PK_A for signer A in a certificateless cryptosystem.

1. Set up the system parameters: First, the KGC (key generator center) selects the master number of k -bits P , where k is the security parameter, and obtains $\{F_p, E / F_p, G, P\}$. Then, $x \in Z_n^*$ is selected as the system master key msk , and the master public key $P_{pub} = xP$ is calculated. Next, hash function are selected: $H_1 : \{0, 1\}^* \rightarrow Z_n^*$, $H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^k$, $H_3 : \{0, 1\}^* \rightarrow \{0, 1\}^k$ and $H_4 : \{0, 1\}^* \rightarrow Z_n^*$. The KGC saves $params = \{F_p, E / F_p, G, P, H_1, H_2, H_3, H_4\}$.

2. Signer A randomly selects $x_A \in Z_n^*$ as the secret value, computes $P_A = x_A \cdot P$, and sends it to the KGC, where the identity of signer A is ID_A . The KGC calculates $h_A = H_1(ID_A, R_A, P_A)$, $R_A = r_A \cdot P$ and $s_A = r_A + h_{Ax} \bmod n$ to generate a partial key $D_A = \{S_A, R_A\}$. Finally, the private key $SK_A = (x_A, s_A)$ and public key $PK_A = (P_A, R_A)$ are produced for signer A in a certificateless public cryptosystem.

Signature-Generation algorithm

The Signature-Generation algorithm generates the signature σ_F for the entire message M . The entire message M is divided into n blocks, $M = \{M[1], M[2], \dots, M[i], \dots, M[n]\}$. M' is any submessage extracted from M . $ext(i)$ represents the number of submessage blocks that are common in M and M' . CEAS is a content extraction access structure. The complete pseudocode for signature generation is given in Algorithm 1.

Algorithm 1 Signature-Generation

Input: $params, ID_A, SK_A, PK_A, M, CEAS$

Output: signature σ_F

```

1: repeat
2:    $c[i] = C(M[i], S_{i_{it}})$ ,  $a[i] = A_n[i] \parallel v_a[i] = c[i]$ 
3: until  $i > n$ 
4:   while  $a \in I_n$  do  $v_a = H_2(v_{a,0}, v_{a,1})$ 
5:   end while
6:   choose  $l \in Z_n^*$ ,  $R = l \cdot P$ 
7:    $h = H_3(v_0, R, PK_A)$ 
8:   if  $\gcd(l + h, n) = 1$  then
9:      $s = (l + h)^{-1} (x_A + s_A) \bmod n$ 
10:   end if  $\sigma_F = (CEAS, R, s, c[i]_{i \in n})$ 
11: return  $\sigma_F$  //generates signature

```

Signcryption-Extraction algorithm

After receiving the signature σ_F , the signcryptor obtains v_0 according to the Signature-Generation algorithm, calculates $h_A = H_1(ID_A, R_A, P_A)$ and $h = H_3(v_0, R, PK_A)$, and verifies the equation $s(R + hP) = P_A + R_A + h_A P_{pub}$. If the equation is not correct, the signcryptor stops the algorithm. Otherwise, the following steps are performed to obtain the signcryption σ_E :

1. Generate $ext(i)$ according to the CEAS.
2. Extract $M' = \{M'[1], M'[2], M'[3], \dots, M'[i], \dots, M'[n]\}$. The specific measures are as follows: if $i \in ext(i)$, $M'[i] = M[i]$, indicating that the submessage block is extracted; Otherwise, $M'[i] = c[i]$.
3. Calculate E_A, E and extract the signcryption σ_E

The complete pseudo code for signcryption extraction is given in Algorithm 2.

Algorithm 2 Signcryption-Extraction

Input: message M , signature σ_F , $ext(i)$, PK_A

Output: signature σ_E

```

1:  $(CEAS, R, s, c[i]_{i \in n}) \leftarrow \sigma_F$ 
2: repeat
3:    $v_a[i] = c[i]$ 
4:   while  $a \in I_n$  do  $v_a = H_2(v_{a,0}, v_{a,1})$ 
5:   end while
6:    $h_A = H_1(ID_A, R_A, P_A)$ ,  $h = H_3(v_0, R, PK_A)$ 
7:   if  $s(R + hP) = P_A + R_A + h_A P_{pub}$  then

```



```

8:     if  $i \in \text{ext}(i)$  then //generates  $\text{ext}(i)$ 
9:        $M[i] = M[i]$ 
10:    else  $M[i] = c[i]$ 
11:    end if
12:  end if
13:   $E_A = l(P_A + R_A + h_A P_{\text{pub}})$ ,  $E = H_4(E_A) \oplus M'$ 
14:   $\sigma_E = (E, \text{CEAS}, \text{ext}(i), R, s)$ 
15:  return  $\sigma_E$ 

```

Signcryption-Verification algorithm

After receiving the signcryption σ_E , the verifier decrypts M' and v_0 and verifies the signcryption by the equation $s(R + hP) = P_A + R_A + h_A P_{\text{pub}}$. The complete pseudocode for signcryption verification is given in Algorithm 3.

Algorithm 3 Signcryption-Verification

Input: submessage M' , signcryption σ_E , PK_A

Output: verification result

```

1:  if  $\text{ext}(i) \in \text{CEAS}$  then //else, stops the algorithm
2:     $E_B = s(x_A + s_A)(P_A + R_A R + h \cdot p)$ 
3:     $M' = E \oplus H_4(E_B)$  //decrypts  $M'$ 
4:  repeat
5:     $c[i] = C(M[i], S_{iR, iC})$ ,  $a[i] = A_n[i]$ ;  $v_a[i] = c[i]$ 
6:    while  $a \in I_n$  do
7:       $v_a = H_2(v_{a,0}, v_{a,1})$  //gets (decrypts)  $v_0$ 
8:    end while
9:     $h_A = H_1(ID_A, R_A, P_A)$ ,  $h = H_4(v_0, R, PK_A)$ 
10:   if  $s(R + hP) = P_A + R_A + h_A P_{\text{pub}}$  then
11:     return Acc //  $\sigma_E$  verified successfully
12:   else
13:     return Rej //  $\sigma_E$  verification failed

```

Correctness analysis

In this section, we prove the correctness by the consistency of the message and the accuracy of the equation.

Message consistency

Message consistency indicates that the extracted message M' in the Signcryption-Extraction algorithm is consistent with the decrypted message M' in the Signcryption-Verification algorithm.

We analyze the consistency between the submessage M'' extracted in the Signcryption-Extraction algorithm and the submessage M''' decrypted in the Signcryption-Verification algorithm.

Submessage M'' is extracted in the Signcryption-Extraction algorithm using the following equation:

$$M''[i] = \begin{cases} M[i], & i \in \text{ext}(i) \\ c[i] = C(M[i], S_{iR, iC}), & i \notin \text{ext}(i) \end{cases} \quad (4)$$

Submessage M'' is decrypted in the Signcryption-Verification algorithm using the following equation:

$$M'''[i] = \begin{cases} c[i] = C(M[i], S_{i_r, i_c}), & i \notin \text{ext}(i) \\ M[i], & i \in \text{ext}(i) \end{cases} \quad (5)$$

The fact is that M'' and M''' are the same, hence, the SECCESPP scheme has consistency.

Equation accuracy

If equation $s(R + hP) = P_A + R_A + h_A P_{pub}$ in the Signcryption-Verification algorithm is true, then the SECCESPP scheme is bounded. In this section, we check $s(R + hP) = P_A + R_A + h_A P_{pub}$ with the following process.

$$\begin{aligned} \because P_{pub} &= x \cdot P, s_A = r + h_A x \bmod n \\ \therefore s_A \cdot P &= R_A + h_A \cdot P_{pub} \\ \because s &= (l + h)^{-1}(x_{ID} + s_{ID}) \bmod n \\ \therefore \text{left} &= s(R + hP) = (h + l)^{-1}(x_A + s_A)(l + h)P \\ &= x_A \cdot P + s_A \cdot P = P_A + R_A + h_A \cdot P_{pub} = \text{right} \end{aligned} \quad (6)$$

Therefore, $s(R + hP) = P_A + R_A + h_A P_{pub}$ is true, and the SECCESPP scheme is bounded.

Security analysis

In this section, first, we demonstrate the security of the SECCESPP scheme under the random oracle model. Then, we use the formal analysis tool ProVerif to formally analyze privacy. Finally, we provide proof of the unforgeability of the SECCESPP scheme.

Security under the random oracle model

The SECCESPP scheme is demonstrably secure under the random oracle model in [35] and can resist adaptive chosen message attack. The possible attacks are divided into two types.

TYPE 1: The attacker does not have access to the primary key. However, the attacker can request or replace the user public key. As discussed above, we impose several natural restrictions on TYPE 1: (1) The attacker cannot extract the private key for ID_i at any point. (2) The attacker cannot request the private key for and identity if the corresponding public has already been replaced.

TYPE 2: The attacker does have access to privacy, but cannot request or replace the user public key. The restrictions on this type of attacker are as follows: (1) The attacker cannot replace public keys at any point. (2) The attacker cannot extract the private key for ID_i at any point.

The proofs for the two types of attacks are similar. Hence, we only present the proof for the attacker who does not access the primary key but can request or replace the user public key.

Definition (ECDLP): For a random number $x \in \mathbb{Z}_n^*$, given two elements P, Q such that $Q = x * P$, the goal of the ECDLP is to calculate x .

Theorem: In the random oracle model, the SECCESPP scheme is secure if the ECDLP is intractable.

Proof: Assume that attacker B who attacks the SECCESPP scheme. Let attacker B construct algorithm F to solve the ECDLP problem.

Initialization Phase: F initializes P and Q and transmits the public parameters to attacker B. The public parameter is $params = \{F_p, E / F_p, G, P, P_{pub} = Q, H_1, H_2, H_3, H_4\}$.

Queries Phase: Attacker B executes the following queries, and F adaptively responds to these queries.

1. **User query:** When attacker B performs a user query on ID_b , challenger F selects a random number $t \in \{1, 2, \dots, q_c\}$. Then, the pseudo-code is executed.

User query pseudo-code

```

1:  if ( $i \neq t$ ) {
2:    F selects  $a, b, c \in Z_n^*$ ,  $s_i \cdot P = R_i + h_i \cdot P_{pub}$ ;
3:    Computes  $R_i = a \cdot P_{pub} + b \cdot P$ ,  $P_i = c \cdot P$ ,  $s_i = b$ ,  $x_i = c$ ,  $h_i = H_1(ID_i, R_i, P_i) \leftarrow -a \bmod n$ ;
4:    Returns  $(ID_i, R_i, P_i, s_i, x_i, h_i)$  and adds into  $L_{H_1}$ ; }
5:  else{
6:    F selects  $a, b, c \in Z_n^*$ ;
7:    Computes  $R_i = a \cdot P_{pub} + b \cdot P$ ,  $P_i = c \cdot P$ ,  $s_i = x_i = \perp$ ,  $h_i = H_1(ID_i, R_i, P_i) \leftarrow -a \bmod n$ ;
8:    Adds  $(ID_i, R_i, P_i, s_i, x_i, h_i)$  to  $C - List$ 
9:    Adds  $(ID_i, R_i, P_i, h_i)$  to  $L_{H_1}$ ;
10:   Returns  $(ID_i, R_i, P_i, h_i)$  to attacker B; }

```

2. **Partial Key Extraction query:** If ID_i does not exist, $\perp$ is output. If ID_i exists and $(i \neq t)$, some of the private keys are returned. Otherwise, F stops.
3. **Public Key Replacement query:** If ID_i does not exist, $\perp$ is output. Otherwise, B replaces user ID_i 's public key PK_i with PK'_i and adds it into L_R .
4. **Secret Value query:** When B executes, it is inputs ID_i . If ID_i does not exist, $\perp$ is output. If ID_i exists and $(i \neq t)$, $x_i = c$ is sent to B. If $(i \neq t)$, then the process stops and returns 'failure'.
5. **Digest Calculation query:** B executes the summary computation query on $\omega_i = (v_0, R, PK_i)$. If ID_i does not exist, $\perp$ is output. Otherwise, for the i -th H_3 query, if $ID_i \neq ID_b$, F queries whether $(ID_i, R_i, P_i, s_i, x_i, h_i)$ exists in $C - List$; If it exists, F picks $h \in Z_n^*$ randomly, and $h = H_4(v_0, R, PK_i)$, whereas if it does not exist, the user's public key has been replaced. In this case, F generates $a, b, c \in Z_n^*$, and lets $s = a$, $R = a^{-1}h_iP_{pub}$ and $h = H_4(v_0, R, PK_i) \leftarrow a^{-1}(r_i + x_i)$. If $ID_i = ID_b$, F picks $\xi \in \{0, 1\}$ at random and sets $\Pr[\xi = 1] = \xi$ and $\Pr[\xi = 0] = 1 - \xi$. If $\xi = 0$, F generates $a, b, c \in Z_n^*$ and $h = H_4(v_0, R, PK_i) \leftarrow a^{-1}(r_i + x_i)$. If $\xi = 1$, F generates $a, b, c \in Z_n^*$ and lets $h = H_4(v_0, R, PK_i)$. Finally, F returns h to B and adds $(ID_i, v_0, R_i, PK_i, h)$ to $C - List$.
6. **Signcryption Extraction query:** When B extracts the signcryption of $(v_0, ID_b, H_4(\omega_i))$, if ID_i does not exist, F outputs $\perp$. When $ID_i \neq ID_b$, if $(ID_i, R_i, P_i, s_i, x_i, h_i)$ exist in $C - List$, F signs it with the corresponding private key. However, if $(ID_i, R_i, P_i, s_i, x_i, h_i)$ does not exist in $C - List$, that means the user's public key has been replaced. Thus, F lets $s = a$, $R = a^{-1}h_iP_{pub}$ and outputs (R, s) as the signcryption. When $ID_i = ID_b$, if $\xi = 1$, F returns 'failure'; otherwise, $\xi = 0$, F lets $s = a$, $R = a^{-1}h_iP_{pub}$, and it outputs (R, s) as the signcryption.

Forgery: B stops the queries and outputs a valid signcryption $(R, s^{(1)})$. If $ID_i \neq ID^*$, 'failure' is declared. Otherwise, the attack is successful. Then, F makes full use of the generalized Forking lemma [36] of certificateless signatures, inputs two different H_2 values, repeats the above process, and obtains two different signatures $(R, s^{(2)})$ and $(R, s^{(3)})$. Then, $s^{(k)}(R + h^{(k)}P) = P_i + R_i + h_iP_{pub}$, $k = 1, 2, 3$. Additionally $R = lP$, $P_i = r_iP$, $P_{pub} = xP$; thus, $s^{(k)}(l + h^{(k)}) = x_i + r_i + h_ix$, $k = 1, 2, 3$.

In the above interrogation process, i , r_i , x are unknown, but F can solve the three unknowns and output x ; therefore, the elliptic curve discrete logarithm problem can be solved.

To solve a given instance of the ECDLP, F is required to successfully execute the following events:

T_1 : F does not stop the whole time.

T_2 : (R, s) is a valid signcryption forgery of (ID^*, PK_{ID}^*, m^*) .

T_3 : q_s is finite and $\xi = 1$.

The probability of the attack of F being successful is

$$Adv_{ECDLP}^F = \Pr[T_1 \wedge T_2 \wedge T_3] = \Pr[T_1] \cdot \Pr[T_2 | T_1] \cdot \Pr[T_3 | T_1 \wedge T_2].$$

Claim1: If T_1 occurs, during the attack of F , the probability of success of the Partial Key Extraction query is $(1 - (1/qc))^{q_{PKE_x}}$, where q_{PKE_x} is the number of times some keys are queried.

Claim2: The probability of success for the Secret Value query is $(1 - (1/qc))^{q_{VE_x}}$, where q_{VE_x} is the number of times the secret value is queried.

Claim3: The probability of success for the Signcryption Extraction query is $(1 - (1/qc)\xi) \geq (1 - \xi)^{q_s}$, where q_s represents the number of times the signcryption is extracted.

As a result, $\Pr[T_1] \geq (1 - (1/qc))^{q_{PKE_x} + q_{VE_x}} (1 - \xi)^{q_s}$. We state that $\Pr[T_2 | T_1] = \epsilon$, so $\Pr[T_3 | T_1 \wedge T_2] = \xi / q_c$. We can obtain $Adv_{ECDLP}^F \geq (1 - (1/qc))^{q_{PKE_x} + q_{VE_x}} (1 - \xi)^{q_s} \cdot (\xi / q_c) \cdot \epsilon$. At $\xi = 1 / (q_s + 1)$, $(1 - \xi)^{q_s} \cdot \xi$ reaches its maximum value, so $Adv_{ECDLP}^F \geq (1 - (1/qc))^{q_{PKE_x} + q_{VE_x}} (1 - 1 / (q_s + 1)) (1 / q_c (q_s + 1)) \epsilon$. Of course, $(1 - (1/qc))^{q_{PKE_x} + q_{VE_x}} (1 - 1 / (q_s + 1)) \times (1 / q_c (q_s + 1)) \epsilon$ is a constant and you cannot ignore ϵ , so you cannot ignore Adv_{ECDLP}^F , which contradicts the hypothesis. The SECCESPP scheme is secure under the random oracle model in [29].

Privacy

We analyze the privacy of a signed message in the SECCESPP scheme using the formal analysis tool ProVerif [37,38]. Privacy is modeled as confidentiality. First, Applied PI is used to formalize the SECCESPP scheme. Then, ProVerif is used for analyzing privacy.

Function and equational theory. The functions and equations used in the modeling process are described in this section. We use the Applied PI calculus to formalize the SECCESPP scheme. Fig 2 depicts the function and equational theory.

The function and equational theory of the SECCESPP scheme mainly includes the public key encryption algorithm $En(x, Pu)$ encrypt the message x with public key Pu and the decryption algorithm $De(y, Pu)$ decrypt the message which the $En(x, Pu)$ encrypted with private key Pu . The function $Pu(y)$ accepts private value y as input and produces public key as output. The function $Pr(y)$ accepts private value y as input and produces private key as output.

Process

The whole process in Fig 3 consists of three processes: the signer process *processSig* in Fig 4, the signcryptor process *processSc* in Fig 5 and the verifier process *processVer* in Fig 6. They constitute the main process together, as shown.

The signer first divides the entire message into n blocks, and then generates a commitment for each block. After that, the signer calculates R and h . Then, the signer compares the value $\gcd()$ and '1'. If they are equal then forwards the signature σ_F to the signcryptor process *processSc* in form of the message m_1 through the public channel c .

$$\left[\begin{array}{l} \text{fun } H_1 / 2. \\ \text{fun } H_2 / 3. \\ \text{fun } H_3 / 3. \\ \text{fun } H_4 / 1. \\ \text{fun } En / 2. \\ \text{fun } De / 2. \\ \text{equation } De(En(x, Pu(y)), Pr(y)) = x \end{array} \right]$$

Fig 2. Function and equational theory.

<https://doi.org/10.1371/journal.pone.0258907.g002>

The signcryptor receives the message m_1 from the signer process through the public channel c . Then it extracts the extracted message blocks and the correspondent commitments. After that, the signcryptor calculates h and h_A . Then it compares the value $s \cdot (R + hP)$ with $P_A + R_A + h_A \cdot P_{pub}$. If they are equal then encrypts the extracted message M' and follows the content extraction signcryption σ_E to the verifier process *processVer* in form of the message m_2 through the public channel c .

The verifier receives message m_2 from the signcryptor process through the public channel c . Then it decrypt M' and $v[i]$, and verifies the signcryption by the equation $s(R + hP) = P_A + R_A + h_A P_{pub}$. Finally, the verifier inputs message m_3 as the verification results.

ProVerif for automatic privacy verification. The privacy of the SECCESPP scheme is modelled as the confidentiality of the signed message M . query attacker: M' is used to model the confidentiality of the signed message. and is added into formal model in Fig 7.

Result analysis. The result in Fig 8 is true, and the SECCESPP scheme has privacy because M' is encrypted before it is sent to the verifier. Attackers can only get encrypted message block, thus, privacy is guaranteed.

Unforgeability

In a certificateless public key cryptosystem, KGC generates a partial private key, and the user generates a secret value and generates a full private key and the public key respectively according to the secret value and the partial private key. This method solves the key management issue in identity-based cryptosystem and the certificate problem in PKI cryptosystem. It is possible for KGC to forge the user's signature because KGC holds part of the user's private key. Therefore, we divide the attack model of the SECCESPP scheme into two types:

process
 (!*processSig* | !*processSc* | !*processVer*)

Fig 3. Main process.

<https://doi.org/10.1371/journal.pone.0258907.g003>

```

[ let processSig= (*Signer process*)
  new i;
  new 1;
  new h;
  new P;
  new M;
  new PKA;
  new SiR,iC;
  let n=len(M) in
  let M=(M[1],M[2],⋮,M[n]) in
  forall i,let c[i]=C(M[i],SiR,iC) in
  ⋮
  let v[i]=c[i] in
  ⋮
  forall v[n],let va=H2(va,0,va,1) in
  let R=1·P in
  let h=H3(v0,R,PKA) in
  if gcd(1+h,n)=1
  then s=(1+h)-1(xA+sA)mod n;
  ^F=(CEAS,R,s,c[i]i∈n)
  let m1=^F
  out(c,m1).
]

```

Fig 4. Process processSig.

<https://doi.org/10.1371/journal.pone.0258907.g004>

Adversary A_I : ordinary user attack. In this attack, the attacker cannot obtain the master key but can replace the public key.

Adversary A_{II} : malicious KGC attack. In this attack, the attacker has the master key and can generate any part of the user's private key, but it is specified that the user's public key cannot be replaced.

Since Adversary A_I type is similar to Adversary A_{II} type but Adversary A_{II} type is more representative, we provide a proof of unforgeability for the Adversary A_{II} type.


```

[let processSc=      (*Signcryptor process*)
new IDA;
new RA;
new PA;
new hA;
new Ppub;
new M';
in(c,m1);
L
let v[i]=c[i] in
L
forall v[i],let va=H2(va,0,va,1) in
let hA=H1(IDA,RA,PA) in
let h=H3(v0,R,PKA) in
if s · (R+hP)=PA+RA+hA · Ppub then
let M'=(M[1],M[2],c[3],M[4]) in
let EA=1 · (PA+RA+hA · Ppub) in
let E=H4(EA) ⊕ M' in
^E=(E,CEAS,ext(i),R,s)
m2=^E
out(c,m2).
]
```

Fig 5. Process processSc.

<https://doi.org/10.1371/journal.pone.0258907.g005>

Theorem: If Adversary A_{II} can output a valid content extraction signcryption σ_E and has not performed a Signcryption Extraction query, then the attacker succeeds, that is, the SECCESPP scheme is broken.

Lemma: If Adversary A_{II} wins the game by at least the probability of ε after q_k User query, q_{PK} Key Extract query and q_s Signcryption Extraction query within a bounded time, then the SECCESPP scheme is said to be unforgeability under an adaptive chosen message attack.

```

[ let processVer= (*verifier process*)
  in(c,m2);
  if CEAS ∈ ext(i), then
    let EB=s(xA+SA)(PA+RA+R+h·p) in
    let M'=E ⊕ H4(EB) in
    forall i, let c[i]=C(M[i],Si) in
    L
    let v[i]=c[i] in
    L
    forall v[n], let va=H2(va,0,va,1) in
    let hA=H1(IDA,RA,PA) in
    let h=H3(v0',R,PKA) in
    if s(R+hP)=PA+RA+hA·Ppub, then
      out(c,m3).
]

```

Fig 6. Process processVer.

<https://doi.org/10.1371/journal.pone.0258907.g006>

Game: The SECCESPP scheme in the Adversary A_{II} case of the adaptive chosen message attack game, which is between challenger C and Adversary A_{II} .

Proof: The security model of unforgeability consist of three phases: Setup Phase, Queries Phase and Forgery Phase. In Queries phase, adversary A_{II} performs multiple queries including User query, Key Extraction query and Signcryption query. challenger C gives corresponding responses.

Setup Phase: Adversary A_{II} makes multiple queries, challenger C maintains lists $l_1 - l_3$ that are empty initially.

Initialization: Challenger C runs the Initialization algorithm. Input security parameter k , challenger C generates $x \in Z_n^*$, system master key P_{pub} and the system parameter $params$, and sends them to Adversary A_{II} . The $params = \{F_p, E / F_p, G, P, P_{pub} = Q, H_1, H_2, H_3, H_4\}$. Set $a \in Z_n^*$.

Queries Phase: Adversary A_{II} executes the following queries, and challenger C adaptively responds to these queries.

1. **User query:** When Adversary A_{II} presents query on ID_b , challenger C maintains a hash list $H_1 - list$ that is initially empty including two-tuples (ID_b, Q_t) . Challenger C checks whether record exists in a hash list $H_1 - list$. If so, challenger C returns corresponding record, else challenger C makes the following responses:

$$Q_t = \begin{cases} (ID_i, R_i, P_i, h_i), i = t \\ (ID_i, R_i, P_i, s_i, x_i, h_i), i \neq t \end{cases} \quad (7)$$

$$\begin{array}{l}
 \text{query attacker: } M' \\
 \\
 \text{let processSig=} \\
 \text{in } (M); \\
 \\
 \text{out}(c, m_1). \\
 \\
 \text{let processSc=} \\
 \text{in}(c, m_1); \\
 \\
 \text{out}(c, m_2). \\
 \\
 \text{let processVer=} \\
 \text{in}(c, m_2); \\
 \\
 \text{out}(c, m_3).
 \end{array}$$

Fig 7. Model for event query.

<https://doi.org/10.1371/journal.pone.0258907.g007>

2. **Key Extract** query: Adversary A_{II} performs queries on the certificateless public key cryptosystem, m_i represents the identity of the user. Challenge C calculates ID_i 's public key PK_i and sends it to Adversary A_{II} .

```

C:\WINDOWS\system32\cmd.exe
-- Query not attacker:M'_10[]
Completing...
Starting query not attacker:M'_10[]
RESULT not attacker:M'_10[] is ture
D:\proverif2.00>

```

Fig 8. The result of the privacy analysis.

<https://doi.org/10.1371/journal.pone.0258907.g008>

3. **Signcryption Extraction** query: Adversary inputs ID_i and a new message M' , and challenger C runs Signcryption-Extraction algorithm according to private key SK_i and sends result to the Adversary A_{II} .

Forgery Phase: Adversary A_{II} inputs $(ID^*, PK^*, M', \sigma^*)$. To win this game, Adversary A_{II} is required to successfully execute the following events:

$T_1: \sigma_E^*$ is a valid signcryption forgery of (ID^*, PK_{ID}^*, M') .

$T_2: ID^*$ never executed Key Extract query.

$T_3: (ID^*, PK_{ID}^*, M')$ no executed Signcryption Extraction query.

Let $Succ_{UF-CMA} = \epsilon = \Pr[T_1 \wedge T_2 \wedge T_3]$ be the probability that Adversary A_{II} wins the game.

Claim1: The probability of success for the Signcryption Extraction query is

$(1 - (1/q_c)^{q_s}) \geq (1 - \xi)^{q_s}$, where q_s represents the number of times the signcryption is extracted and $\xi \in \{0, 1\}$.

Claim2: The probability of success for the User query is $(1 - (1/q_c))^{q_k}$, where q_k is the number of times the private key is queried.

Claim3: During the attack of Adversary A_{II} , the probability of success for the Key Extract query is $(1 - (1/q_c))^{q_{PK}}$, where q_{PK} is the number of times public key are queried.

As a result, $\Pr[T_1] \geq (1 - (1/q_c))^{q_{PK} + q_k} (1 - \xi)^{q_s}$. We say that $\Pr[T_2 | T_1] = \epsilon$, so $\Pr[T_3 | T_1 \wedge T_2] = \xi / q_c$. We can obtain $Succ_{UF-CMA} \geq (1 - (1/q_c))^{q_{PK} + q_k} (1 - \xi)^{q_s} \cdot (\xi / q_c) \cdot \epsilon$. At $\xi = 1 / (q_s + 1)$, $(1 - \xi)^{q_s} \cdot \xi$ reaches its maximum value, so $Succ_{UF-CMA} \geq (1 - (1/q_c))^{q_{PK} + q_k} (1 - 1/(q_s + 1))^{q_s} (1/q_c(q_s + 1)) \epsilon$. Of course, $(1 - (1/q_c))^{q_{PK} + q_k} (1 - 1/(q_s + 1))^{q_s} (1/q_c(q_s + 1)) \epsilon$ is a constant and you cannot ignore ϵ , so you cannot ignore $Succ_{UF-CMA}(\epsilon)$, which satisfies what Lemma says. Therefore, the SECCESPP scheme has unforgeability.

Comparison and discussion

In the SECCESPP scheme, we use scalar multiplication on elliptic curves, thereby reducing the number of calculations in the signing and verification process. For efficiency, we compare two aspects of the proposed scheme to those of related schemes in [14–19]. One is the theory calculation aspect. The other is the practical running time aspect. For the theory calculation aspect, the SECCESPP scheme is compared in terms of the following four factors in Tale 1: exponential operation (exp), scalar multiplication (sca), bilinear pairing operation (par) and hash function (has), where n is the number of submessages, m is the number of the extracted messages, m_{CEAS} is the number of submessages in the content extraction access structure CEAS. The SECCESPP scheme has “ $(2n+2)has+4sca$ ” and “ $(m+2)has+2sca$ ” calculation, which are the lowest amounts. Hence, the SECCESPP scheme is highly efficient from the theoretical calculation aspect.

Table 1. The number of operation.

scheme	Signcryption and extraction	Verification
[14]scheme	$(2n+3)\text{has}+2\text{exp}+1\text{par}+4\text{sca}$	$(m+3)\text{has}+1\text{par}+2\text{sca}$
[15]scheme	$(2n+3)\text{has}+7\text{exp}$	$(m+3)\text{has}+5\text{exp}$
[16]scheme	$(2n+3)\text{has}+5\text{exp}$	$(m+3)\text{has}+3\text{exp}$
[17]scheme	$(2n+3)\text{has}+4\text{sca}$	$(m+3)\text{has}+3\text{sca}$
[18]scheme	$(5n)\text{sca}$	$(3m)\text{par}$
[19]scheme	$(2n+2)\text{exp}+1\text{par}$	$(m+3)\text{par}+2\text{exp}$
our scheme	$(2n+2)\text{has}+4\text{sca}$	$(m+2)\text{has}+2\text{sca}$

<https://doi.org/10.1371/journal.pone.0258907.t001>

Table 2. Computing time on operations.

Operations	has	exp	par	sca
running time	2.01ms	3.31ms	10.02ms	1.51ms

<https://doi.org/10.1371/journal.pone.0258907.t002>

For the practical running time aspect, the hardware platform consists of Intel Core m3-6Y30 @0.90 GHz processor and 8GB memory. The software environment includes Windows 10 operating system for 64 bits and Miracl library [39] for which the parameters are specified as follows: the supersingular elliptic curve $E / F_p: y^2 = x^3 + x$ is selected, in which the embedding degree is 2 and the prime number p satisfies $2^{510} < p < 2^{511}$, $p + 1 = 12qr$, $q = 2^{159} + 2^{17} + 1$. The Tate pair operation is defined in $E / F_p: y^2 = x^3 + x$. The scalar multiplication operations on the elliptic curves satisfy $p = 2^{160} - 2^{31} - 1$.

To avoid randomness, the simulation experiments are performed five times to obtain the averaged results. The computing time of the four factors is shown in Table 2. According to Tables 1 and 2, the running time in Table 3 are calculated.

The running time of the scheme depends on the signcryption and extraction time and verification time. From Table 3 and Fig 9, we can clearly see that the scalar multiplication operation on the elliptic curves takes much less time than bilinear pair operation and exponential operation. Hence the analysis result of theory calculation is consistent to analysis result of the practical running time. Therefore, the SECCESPP scheme has higher efficiency than compared scheme.

Table 3. Running time.

scheme	Signcryption and extraction/ms	Verification/ms
[14] scheme	$4.02n+28.71$	$2.01m+19.07$
[15] scheme	$4.02n+29.20$	$2.01m+22.58$
[16] scheme	$4.02n+22.58$	$2.01m+15.96$
[17] scheme	$4.02n+12.07$	$2.01m+10.56$
[18] scheme	$7.55n$	$30.06m$
[19] scheme	$6.62n+16.64$	$10.02m+36.68$
our scheme	$4.02n+10.06$	$2.01m+7.04$

<https://doi.org/10.1371/journal.pone.0258907.t003>

Fig 9. Comparison of running time in the signing and verifying process.

<https://doi.org/10.1371/journal.pone.0258907.g009>

Conclusion

To improve the efficiency of and provide privacy protection for content extraction signatures, we proposed the SECCESPP scheme in which the scalar multiplication on elliptic curves is used to replace inefficient bilinear pairing in a certificateless public key cryptosystem, and the signcryption idea is borrowed to provide privacy protection. The SECCESPP scheme is provably secure based on the elliptic curve discrete logarithm problem in the random oracle model. It not only has correctness and privacy, but is also more efficient than related schemes [14–19].

In the future, we will use the SECCESPP scheme to the off-chain data access in blockchain to implement security and privacy.

Author Contributions

Conceptualization: Jiabing Liu.

Investigation: Fuyong Zheng.

Validation: Dejun Wang.

Visualization: Bo Meng.

Writing – original draft: Can Zhao.

References

1. Azees M., Vijayakumar P. et al., “An efficient anonymous authentication and confidentiality preservation schemes for secure communications in wireless body area networks”. *Wireless Netw.*, Vol. 27, pp. 2119–2130, 2021.
2. Y. Zheng. “Digital signcryption or how to achieve $\text{cost}(\text{signature} \& \text{encryption}) \ll \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$ ” in *Advances in Cryptology—CRYPTO ’97*, Lecture Notes in Comput. Sci. Vol. 1294, Springer, Berlin, 1997, pp. 165–179.
3. Ma J.H., Liu J.H., Wu W., Huang X.Y., “Survey on Redactable Signatures”. *J. of Computer Research and Development*, Vol. 54, no.10, pp.2144–2152, 2017.
4. Li X., Du X.N., Wang C.F. “Improved scheme of content extraction signatures based on RSA”. *Computer Engineering and Applications*, Vol. 50, no. 24, pp.96–99, 2014.
5. Yan J. H., Wang L, Wang L.C., et al. “Attribute-based signcryption from lattices in the standard model”. *IEEE Access*, Vol. 2019, no. 7, pp. 56039–56050.
6. J. Malone-Lee. “Identity-based signcryption” in *Cryptology ePrint Archive*, 2002. <http://eprint.iacr.org/>.
7. X. Boyen. “Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography” in *Advances in Cryptology—CRYPTO 2003*, Vol. 2729 of LNCS, pp. 382–398. Springer-Verlag, 2003.
8. Wang X., Zhang Y., Gupta B.B., et al. “An Identity-Based Signcryption on Lattice without Trapdoor”. *J. UCS*, Vol. 25, no. 3, pp.282–293, 2019.
9. Shankar K., Elhoseny M., Perumal E., et al. “An Efficient Image Encryption Scheme Based on Signcryption Technique with Adaptive Elephant Herding Optimization”. *Cybersecurity and Secure Information Systems*. Springer, Cham, 2019, pp. 31–42.
10. D. Pan, B. Liang, H. Li, et al. “Hierarchical Functional Signcryption: Notion and Construction”. *International Conference on Provable Security*. Springer, Cham, 2019, pp. 167–185.
11. M. Barbosa, P. Farshim. “Certificateless signcryption” in *ACM Symposium on Information, Computer and Communications Security-ASIACCS 2008*. Tokyo, Japan, 2008, pp. 369–372.
12. Gao G.M., Peng X.G., Qin Z.F. “Certificateless Signcryption Scheme without Bilinear Pairing” *J. of North University of China (Natural Science Edition)*. Vol. 40, no. 02, pp. 118–125, 2019.
13. Wang C.F., Xu Q.B., Liu C., et al. “Partial Blind Signcryption Scheme in CLPKC-to-TPKI Heterogeneous Environment”. *J. of Electronics & Information Technology*. Vol. 41, no. 8, pp. 1823–1830, 2019.
14. C. Wang, Y. Li, S.Y. Huang, et al. “A new forward secure content extraction signature scheme” in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2015, pp. 1698–1702.

15. Thirumalai C. "Review on the memory efficient RSA variants". *International Journal of Pharmacy and Technology*, Vol. 8, no. 4, pp. 4907–4916, 2016.
16. T.B. Idalino, L. Moura, C. Adams. "Modification Tolerant Signature Schemes: Location and Correction" in *International Conference on Cryptology*. India. Springer, Cham, pp. 2019: 23–44.
17. Kruliš M., Lokoč J., Skopal T. "Efficient extraction of clustering-based feature signatures using GPU architectures". *Multimedia Tools and Applications*, Vol. 75, no. 13, pp. 8071–8103, 2016.
18. Karati A., Fan C., Hsu R. "Provably Secure and Generalized Signcryption With Public Verifiability for Secure Data Transmission Between Resource-Constrained IoT Devices" in *IEEE Internet of Things Journal*, Vol. 6, no. 6, pp. 10431–10440, 2019.
19. Shim K. "A New Certificateless Signature Scheme Provably Secure in the Standard Model" in *IEEE Systems Journal*, Vol. 13, no. 2, pp. 1421–1430, 2018.
20. Stergiou C.L., Psannis K. E. and Gupta B.B., "IoT-Based Big Data Secure Management in the Fog Over a 6G Wireless Network" in *IEEE Internet of Things Journal*, Vol.8, no. 7, pp. 5164–5171, April. 2021.
21. Ye Q.Q., Meng X.F., Zhu M.J. "Survey on Local Differential Privacy". *J. of Software*, Vol.29, no.7, pp. 159–183, 2018.
22. Stergiou C.L., Psannis K. E. and Gupta B.B., "IoT-Based Big Data Secure Management in the Fog Over a 6G Wireless Network" in *IEEE Internet of Things Journal*, Vol.8, no. 7, pp. 5164–5171, April. 2021.
23. Zhao X.F., Fu Y., Song W.T. "Circular Secure Homomorphic Encryption Scheme". *J. of Computer Research and Development*, Vol.57, no.10, pp. 2117–2124, 2020.
24. Zhang J. et al, "A Secure Decentralized Spatial Crowdsourcing Scheme for 6G-Enabled Network in Box" in *IEEE Transactions on Industrial Informatics*, 2021, pp. 99:1–1.
25. Vijayakumar P., Azees M., Kozlov S. and Rodrigues J. J. P. C., "An Anonymous Batch Authentication and Key Exchange Protocols for 6G Enabled VANETs" in *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2021.
26. Azees M., Vijayakumar P., et al, "BBAAS: Blockchain-Based Anonymous Authentication Scheme for Providing Secure Communication in VAENTs". *Security and Communication Networks*, Vol. 2021, pp. 11–23, 2021.
27. R. Steinfeld, L. Bull, Y. Zheng. "Content Extraction Signatures" in *International Conference on Information Security and Cryptology ICISC 2001*, Vol. 2288, Springer-Verlag, Berlin, 2001, pp. 285–304.
28. Lan C.H., Wang C.F. "ID-based content extraction signature". *J. of Computer Applications*, Vol.27, no. 10, pp.2456–2458, 2007.
29. Wang M., Ma J.H., Ma J.H. "Two improved content extraction signature schemes". *Chinese J. of Network and Information Security*, Vol.2017, no. 04, pp. 73–81.
30. Thirumalai C. "Review on the memory efficient RSA variants". *International J. of Pharmacy and Technology*, Vol. 8, no. 4, pp. 4907–4916, 2016.
31. P. Szalachowski. "A Blockchain-Based TLS Notary Service" in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019, pp. 836–843.
32. Cheng L.J., Qi Z.H., Shi J.C. "Blockchain based secure storage and sharing scheme for HER data". *J. of Nanjing University of Posts and Telecommunications*, Vol. 40, no. 04, pp. 96–102, 2020.
33. Fang G.W., Cao J.X. "Provably secure certificateless multi-message synchronous broadcast signcryption scheme". *J. of Southeast University*, Vol. 50, no. 04, pp. 728–740, 2020.
34. Deng F., Wang Y., Peng L., et al. "Revocable Cloud-Assisted Attribute-Based Signcryption in Personal Health System". *IEEE Access*, Vol. 2019, no. 7, pp. 120950–120960.
35. S.S. Al-Riyami, K.G. Paterson. "Certificateless Public Key Cryptography" in Lai H. CS. (eds) *Advances in Cryptology—ASIACRYPT 2003*. ASIACRYPT 2003. Lecture Notes in Computer Science. Vol. 2894. Springer, Berlin, Heidelberg.
36. David P., Jacques S. "Security Arguments for Digital Signatures and Blind Signatures". *J. of Cryptology*, Vol. 13, no. 3, pp. 361–369, 2000.
37. Liu J., Wang L. and Yu Y., "Improved Security of a Pairing-Free Certificateless Aggregate Signature in Healthcare Wireless Medical Sensor Networks" in *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5256–5266, June 2020.
38. Abbasinezhad-Mood D. and Nikooghadam M., "Efficient Anonymous Password-Authenticated Key Exchange Protocol to Read Isolated Smart Meters by Utilization of Extended Chebyshev Chaotic Maps" in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4815–4828, Nov. 2018.
39. "Miracl: Multiprecision integer and rational arithmetic C/C++ library", [Online]. <http://indigo.ie/msott/>.