Efficient and Revocable Anonymous Account Guarantee System Based on Blockchain

Weiyou Liang[1], Yujue Wang[2], Yong Ding[1,4], Hai Liang[1(✉)], Changsong Yang[1], and Huiyong Wang[3]

[1] Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China
lianghai@guet.edu.cn

[2] Hangzhou Innovation Institute of Beihang University, Hangzhou, China

[3] School of Mathematics and Computing Science, GuilinUniversity of Electronic Technology, Guilin, China

[4] Institute of Cyberspace Technology, HKCT Institute for Higher Education, Hong Kong SAR, China

Abstract. The fast expansion of information technology and public concern for personal privacy and security have raised expectations for the authentication process. Although existing anonymous authentication schemes can achieve anonymous authentication and accountability, they all require users to apply for certificates from the authorization authority, resulting in a significant certificate storage overhead for the authority. Additionally, they have not implemented certificate revocation for anonymous users, which allows malicious users to potentially engage in malicious behavior. Therefore, this paper proposes an efficient and revocable anonymous account guarantee system based on blockchain (ERAAS). The system implements a guarantor mechanism where anonymous users can authenticate their identities through the guarantees provided by guarantors without the need to apply for certificates, reducing the storage overhead of certificates. Furthermore, the system utilizes cryptographic accumulators to enable fast revocation of accounts, preventing malicious users from engaging in further malicious behavior. Moreover, in this system, the certificate authority (CA) can enhance the system's ability to handle concurrent requests by allocating group keys to the registration authority (RA), authorizing them to register guarantors and sign guarantees. Security analysis indicates that the proposed scheme enjoys anonymity, traceability, and revocability and can resist forgery attacks. The experimental comparison demonstrates its practicality.

Keywords: Authentication · Anonymity · Revocable · Supervision · Blockchain

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 561, pp. 167–186, 2024.
https://doi.org/10.1007/978-3-031-54521-4_10

1 Introduction

With the fast growth of information technology, more and more individuals are used to working, buying, and socializing online. The Internet, like anything else, has both advantages and disadvantages. On one hand, its emergence has brought convenience to people's lives, such as online signing of multi-party electronic contracts [27], online chatting, and more, effectively removing geographical limitations for humanity. However, its popularity is accompanied by security worries [12]. One of them is the illegal access of malevolent users to valuable data or services. In applications like smart grids, if data becomes accessible to malicious actors, it can lead to issues such as user privacy breaches [4]. In order to lessen the likelihood of these security events, researchers have performed a significant amount of work on identity authentication systems. Identity identification technology research is still a prominent topic today.

In the online world, authentication is essential for safeguarding personal and other sensitive data from unauthorized access by third parties, preventing users from participating in improper online behavior, and ensuring that a computer is not hacked or infected with a harmful virus [28]. The most prevalent identity authentication tool in the realm of identity authentication technology is the digital certificate. It is a cryptographic system that functions as the Internet's passport. After the verifier has the digital certificate offered by the presenter, using the public key and private key information it possesses, it may then determine whether the communication originated from a particular individual [13]. Of course, digital certificates have a relatively large privacy leakage problem because the presenter needs to expose all her attributes when presenting the certificate. This may not seem important, but it is. Wherever you least expect it, a breach of your personal information might occur. According to the Identity Theft Resource Center, the number of hacked records holding sensitive personally identifiable information (PII) is growing. PII is any information that may be used to identify or track a person [19]. Hence, we require a novel authentication mechanism capable of both confirming identity like digital certificates and concealing individual identities.

Consequently, the researchers came up with the concept of anonymous authentication. It is possible for a user who is required to verify herself to do so without disclosing her identity. This makes it possible to limit access while still protecting the privacy of the user. For applications like blockchain, which prioritize privacy protection, anonymous authentication plays a significant role in ensuring security [24]. Currently, anonymous authentication techniques may be loosely categorized as schemes based on public key cryptosystems, schemes of cryptosystems based on identity, schemes based on pseudonyms, and mixed schemes [17]. IBM's identity mixer scheme, which allows users to selectively present their own identity attributes, is one of the most prominent anonymous authentication schemes. Other people cannot determine her specific identity even if they see the certificate she presents, and they cannot link the anonymous certificates she presented multiple times [3].

Even though an anonymous authentication method, such as the identity mixer technique, may accomplish identity authentication and safeguard user identity privacy, it will be susceptible to anonymity abuse owing to the absence of a supervisor. Therefore, we must employ regulators to uncover and penalize bad behavior by anonymous users [25]. Liang et al. [14,15] suggested a double-layer structure for CA supervision that assures the efficiency of supervision and certificate issuing. During the authentication process, users can undergo anonymous authentication without disclosing their full identity attributes. When a user behaves maliciously, the CA may identify an anonymous user by utilizing the supervisory private key. However, like this kind of scheme, the user must register with the CA and apply for a certificate to accomplish anonymous authentication. In this situation, CA's storage overhead is high. Cheng et al. [5] presented a technique that may conduct anonymous authentication, monitor, and decrease certificate storage. There may be two main issues with this scheme in real-world applications. In the original method, there is just one CA. Thus, when a significant number of guarantee requests are sent in a short period of time, the CA's processing efficiency becomes problematic. Also, the previous strategy merely oversaw the guarantor but did not account for the deactivation of the harmful anonymous account, so it may request services again.

1.1 Our Contributions

This paper proposes an ERAAS system based on blockchain that not only achieves anonymous authentication but also incorporates accountability and revocation mechanisms. The system aims to prevent the misuse of anonymity and the possibility of malicious users engaging in subsequent illegal activities. The specific contributions of this system are outlined below.

1. The ERAAS system implements anonymous and controllable authentication, allowing legitimate users to authenticate without revealing personal information. For malicious behavior by anonymous users, trusted authorities can trace the guarantor's real identity through evidence chains, preventing abuse. Security and usability are confirmed through analysis and experiments.
2. In ERAAS, guarantors can use their polynomial to generate account guarantees, allowing users to authenticate without authority-issued certificates, reducing storage overhead for the authority.
3. The ERAAS system implements a revocation mechanism to prevent malicious users from reusing their accounts for authentication after removal from the accumulator, effectively deterring future malicious activities. This mechanism offers fast revocation and consistent additional authentication time, independent of the number of revocations, making it a fixed overhead cost.
4. The ERAAS system also proposes to authorize RAs using group signature technology so that RAs can issue guarantee certificates for guarantors and sign for anonymous account guarantees, thus improving the system's ability to handle concurrent requests.

1.2 Related Works

Over the course of the last several decades, user authentication systems have grown more widespread in the activities that we participate in a daily basis. Their major objective is to effectively stop unauthorized users from accessing sensitive data and services [22]. The digital certificate, which associates a user's identifying information with a specific collection of data, is the identity authentication technique that is used the most often nowadays. In 1988, ITU-T established the standard for digital certificates and released the first version of X.509 [9]. After that, digital certificates became a key method of identity identification, and the standard for them was established by ITU-T. Even up to the present day, there are still a significant number of researchers working at X.509. Zulfiqar et al. [29] evaluated the revocation processes used by the highest-ranking websites. Their data indicates that the adoption of the online certificate status protocol (OCSP) has been much slower than increases in public key strength and sequence number randomization. However, the use of OCSP facilitates the rapid validation of X.509 certificates and the rejection of expired certificates. Saleem et al. [20] proposed a framework named ProofChain. It is a decentralized public key infrastructure (PKI) solution that was introduced by Saleem et al. and is built on the blockchain. It facilitates complete trust across decentralized CA groups. Although a digital certificate such as X.509 may accomplish identity verification, because it requires the presenter to demonstrate her identity attributes, the identity privacy of the user is compromised in this instance. Therefore, individuals need a new authentication technique that can both authenticate their identities and secure their privacy.

In order to protect user privacy and complete identity authentication at the same time, researchers have conducted a lot of research. Gao et al. [6] proposed a proxy mobile IPv6-based anonymous authentication mechanism for vehicular ad hoc networks (VANETs). The method is equipped with pseudonyms, identity-based password procedures, and a number of crucial authentication protocols that protect the anonymity of users. Liu et al. [16] designed an anonymous authentication technique that does not need tamper-resistant devices using lattice-based encryption. For anonymous vehicle authentication, this technique may produce a fictitious identifier for each vehicle. In addition, the technique is capable of preventing channel-testing assaults. Wang, Xu and Gu [21] suggested an enhanced authentication technique based on the cryptography of elliptic curves. Their approach can withstand offline password guessing attacks, desynchronization assaults, and session key disclosure attacks, while also achieving anonymous user authentication to safeguard user privacy.

Banerjee et al. [2] introduced an anonymous authentication technique that enables users to connect across insecure communication channels while protecting critical information. This strategy provides demonstrable security. It is more secure and resilient compared to prior methods. Han et al. [7] proposed an anonymous authentication scheme for VANETs based on fog computing, which can protect the privacy of vehicles. This scheme designs a pseudonym update and tracking approach that is based on fog computing. This strategy has the poten-

tial to minimize the amount of time it takes to authenticate valid cars, which in turn will increase overall efficiency. At the same time, the system makes use of self-authentication, which lessens the burden of communication placed on the center and enhances the efficiency of the authentication process.

Although anonymous authentication can protect the privacy of users, complete anonymous authentication will lead to the abuse of anonymity, so regulators need to carry out supervision. Based on the rotating group signature scheme of elliptic curve cryptography, Mehmood et al. [17] proposed an anonymous authentication scheme for healthcare applications based on intelligent clouds. When using an untrusted authentication server, this approach may ensure the user's anonymity and avoid eavesdropping. In addition, it offers a tool for tracking to prevent anonymous misuse. Jegadeesan et al. [10] proposed a safe and efficient privacy-preserving anonymous authentication scheme, which anonymously verifies users with less computational cost. At the same time, it can also disclose the actual identity of the misbehaving doctor at a very low computational cost, avoiding the situation where the doctor's anonymity is abused. Jiang, Ge and Shen [11] proposed a method for anonymous authentication based on the group signature scheme, which enables anonymous authentication of automobiles with adjacent vehicles or roadside infrastructures. With the private key, the authority can trace an anonymous user's real identity during harmful incidents.

Arasan et al. [1] developed a computationally efficient and more secure anonymous authentication technique for cloud users, which may enable anonymous mutual authentication between cloud users and cloud servers, safeguarding the privacy of users and servers. The scheme's suggested condition-tracking method may guarantee that a trustworthy third party can remove the cloud user's or service provider's access to the cloud environment in the event of harmful activity. Zhang et al. [26] suggested an anonymous authentication scheme for batch verification based on elliptic curves, which may significantly increase the authentication efficiency of the system since it enables batch verification. At the same time, the implementation of anonymous authentication that it provided may help secure users' personal information. Naturally, in the event that anonymous users engage in malevolent behavior, illegal cars may also be traced in order to provide conditional privacy protection. Cheng et al. [5] proposed an anonymous account guarantee scheme based on polynomials. This method guarantees the legitimacy of anonymous accounts. A legitimate user can guarantee multiple anonymous accounts, which can reduce the storage overhead of trusted centers to store certificates.

2 System Model and Security Requirements

2.1 System Model

As shown in Fig. 1, an ERAAS system consists of six types of entities.

– CA: It is a trusted participant in the entire system, responsible for issuing certificates to RA, enabling RA to have the authority to issue certificates to

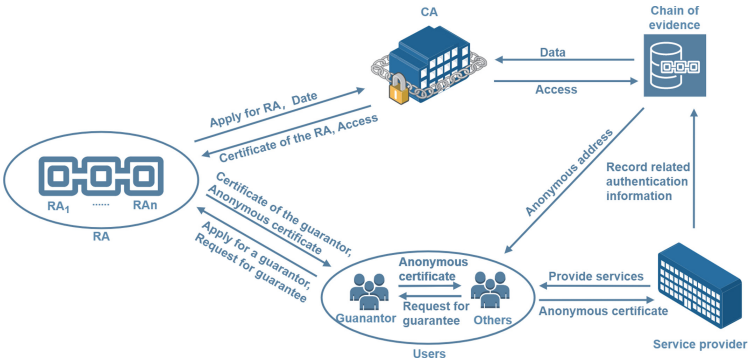

Fig. 1. The ERAAS system model.

guarantors and sign guarantees for ordinary users. At the same time, when there is a dispute between a service provider and a malicious user, CA can trace the real identity of the guarantor and revoke the malicious account.

- RA: It is an organization recognized by CA that can issue certificates to guarantors and sign guarantees for users. Its main responsibility is to issue certificates to guarantors and sign guarantees for ordinary users, and it can also trace the real identity of the guarantor.
- Guarantor: A user who has qualified guarantee status and possesses a certificate issued by the RA. The guarantor can use their credentials to guarantee the anonymous accounts of both themselves and other ordinary users.
- Others: They are ordinary users without guaranteed capability. Before undergoing anonymous identity authentication, they need to find a guarantor to guarantee their own anonymous account.
- Service provider: A trusted service provider who has access to the system. After successfully verifying the anonymous identity presented by the user, the service provider provides services to the user and broadcasts the corresponding record to the entire network, generating a corresponding block to record relevant information. Otherwise, the service provider will refuse to provide services to the user.
- Chain of evidence: The evidence chain leverages blockchain's tamper-evident properties for evidence preservation. Service providers broadcast user-specific service details, anonymous certificates, and related information to the network, storing them on the blockchain. In case of disputes needing arbitration, the CA traces the guarantor's true identity via certificates on the chain.

After the establishment of the ERAAS system, the CA confirms the highest degree of the polynomial and the maximum value of the coefficient, generates a tracking key and a group public key, and initializes the value of the accumulator. Then, during the authorization stage, RA applies for registration with CA, and CA assigns group member private keys to RA. CA sends the private keys of the group members and the accumulator through a secure channel to RA.

Following verification of RA's identity during the guarantor registration stage, the guarantor applies for registration with RA using the real identity and an identity tag *pse*. The RA receives the registration request from the guarantor and verifies their real identity. If passed, RA assigns the guarantor a unique polynomial and stores the actual identity information along with the polynomial.

A user who requires an anonymous account guarantee finds a guarantor to offer a guarantee at the guarantee application stage. Based on the user's private address, the guarantor figures out the guarantee value and sends it to RA to be checked and signed. The anonymous identity can be used for authentication once it has been guaranteed by the guarantor and signed by the RA.

During the service request stage, the user only needs to provide the service provider with their anonymous account and the guarantor's signature. After verification, the service provider provides the service. The service provider records the user's anonymized address, signature, and obtained services on the blockchain once the service is concluded.

During the supervision stage, the service provider provides the CA with the record's block hash value, and the CA uses its tracking key to locate the RA. Then, RA investigates the guarantor's real identity. In addition, the CA can revoke the pernicious account's address to prevent it from engaging in malicious activities in the future.

2.2 Security Requirements

A secure ERAAS system should satisfy the following requirements.

- Anonymity: Without access to the identity information of the guarantor, an adversary cannot guess the true identity of the guarantor of the certificate presented by an honest user during anonymous authentication.
- Traceability: An adversary can't collude with the RA for an untraceable signature or recover the guarantor's polynomial from an anonymous account's guarantee value, preventing them from forging a valid anonymous certificate.
- Revocability: After an anonymous account exhibits malicious behavior, the CA should be able to revoke the account to ensure that the malicious account cannot pass identity authentication again.
- Anti-forgery attack: Without access to the private key of RA and the polynomial of the guarantor, an adversary cannot forge a valid certificate.

3 ERAAS Construction

This paper designs an ERAAS system based on blockchain. The process of the whole system is as follows.

3.1 System Setup (Setup)

The CA selects a large prime number p, and confirms the format of polynomial functions, such as equation (1), where the highest power of the polynomial

function is $n \in Z_p$, and polynomial coefficients are $a_0, \cdots a_n \in Z_p$. Assuming a polynomial function represents a guarantor, the number of guarantors they can represent is shown in the Table 1.

$$f(x) = a_n x^n + a_{(n-1)} x^{n-1} + \cdots + a_1 x + a_0 \tag{1}$$

Table 1. The number of guarantors

	$n = 3$	$n = 4$	$n = 5$	$n = 6$
$a = 2^5$	2^{15}	2^{20}	2^{25}	2^{30}
$a = 2^{10}$	2^{30}	2^{40}	2^{50}	2^{60}
$a = 2^{15}$	2^{45}	2^{60}	2^{75}	2^{80}

Thus, the system does not need to use large coefficients or high power polynomial functions to distinguish all guarantors. Then it chooses two bilinear cyclic groups $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$, satisfying bilinear pairing $e : G_1 \times G_2 \rightarrow G_T$. CA randomly selects $g_3 \in G_1$ and a collision-resistant hash function $H : \{0,1\}^* \rightarrow Z_p$. Finally, it outputs parameters $Para(f(x), n, G_1, G_1, G_T, e, g_1, g_2, g_3, g_4, g_5, p, H)$.

3.2 Generate Key (Genkey)

The CA randomly selects $d, s, t \in {Z_p}^*$, and $D = g_1^d, S = g_2^s$, and $T = g_1^t$. Thus, the CA gets the secret group key $GSK = (d, s, t)$, and the public group key $GPK = (D, S, T)$. Besides, it randomly selects $z, v \in {Z_p}^*$ and computes $L = g_3^l$, $Acc_0 = g_2^v$. Then, it outputs the public key $PK(D, S, T, L, v, (g_3^l, g_3^{l^2}, \cdots, g_3^{l^q}))$.

3.3 RA Registration (RReg)

If the RA_i wants to get the power to register for the guarantor and the ability to issue a certificate, it needs to use its physical identity to register with the CA, then the RA_i will get a private key. The specific process of the registration described is as follows.

1. The CA randomly selects an element $x_i \in Z_p^*$ and calculates

$$R_i = g_1^{(d-x_i)(s*x_i)^{-1}} \tag{2}$$

   Since $R_i = g_1^{r_i}$, we know $x_i + r_i * s * x_i = d$.
2. The CA checks whether x_i is different from the previous ones. If not, the CA will repeat the previous step.
3. The private key RSK_i of the registration center is x_i. The CA stores the corresponding identity information and $R_i^{x_i}$.

CA sends x_i and l to the RA_i through a secure connection.

3.4 Guarantor Registration (GReg)

It is not necessary for all users to complete the phase. The guarantor must be aware that once she guarantees her own or another person's account, she will be responsible for the actions of the anonymous account. Therefore, the guarantor should not guarantee an unknown person's anonymous account.

During the first step of the guarantor registration procedure, the guarantor will send a message m_r to RA_i through a secure channel to verify its identity. When the RA_i receives a message m_r from a new guarantor, it will sign the message m_r with its group's private key so that the guarantor can authenticate its legal identity. The specific signature process is described as follows.

1. The RA_i randomly selects $k_a \in Z_p^*$ and calculates auxiliary values as follows.

$$A_1 = g_1^{k_a} \tag{3}$$

$$A_2 = R_i^{x_i} \cdot T^{k_a} \tag{4}$$

$$Q_1 = e(T, S)^{k_a} \tag{5}$$

2. The RA_i calculates the challenge value as follows.

$$C_{au} = H(m_r, A_1, A_2, Q_1) \tag{6}$$

3. Then RA_i calculates $w_a = k_a * C_{au} + x_i$. It outputs the signature as follows.

$$Sig := (A_1, A_2, C_{au}, w_a) \tag{7}$$

Once the guarantor has RA_i's signature, she will perform identity verification utilizing the group public keys and information m_r. The following are the specific steps that make up the verification process.

1. The guarantor calculates the auxiliary value $\tilde{Q}_1$ as follows.

$$\tilde{Q}_1 = \frac{e\left(A_2, S\right) \cdot e(g_1, g_2)^{w_a}}{e(A_1^{C_{au}} \cdot D, g_2)} \tag{8}$$

2. The guarantor calculates the challenge value.

$$C'_{au} = H(m_r, A_1, A_2, \tilde{Q}_1) \tag{9}$$

3. Finally, the guarantor checks whether $C'_{au} = C_{au}$ holds. If true, the guarantor accepts Sig and registers with the RA_i. Otherwise, she rejects Sig.

After the guarantor accepts the signature, she registers with the RA_i. To differentiate herself from the other users, the guarantor is expected to choose a unique pseudonym pse and generate an ID to serve as verification of her identity. After that, she selects a polynomial function $f()$ at random, making sure that it is compliant with the format for polynomials that was established during the setup phase. This function necessitates the use of the highest possible power, denoted by n, as well as a finite field, which is denoted by the prime element p.

Then, she sends RA_i a registration request *regreq* across a secure connection, and the request will include the following information.

$$regreq := (pse, ID, f()) \tag{10}$$

When the RA_i receives a registration request from a new guarantor, it is required to check certain components of the request. First, it is necessary for RA_i to verify the guarantor's actual identity through ID. Second, the RA_i has to make sure that the guarantor has not already received certificates on the blockchain network while using the same pseudonym *pse*. After that, the RA_i verifies that the guarantor's request contains a polynomial function in the correct format. In the fourth step, the RA_i is responsible for ensuring that the polynomial function $f()$ inside the blockchain network is unique. At last, the RA_i will upload the guarantor's registration request to the blockchain. After the preceding steps, the RA_i provides the guarantor with a registration response. In the event that any of the above steps are unsuccessful, the guarantor's registration will be invalid, she will be informed of her failure, she has to redo the guarantor registration phase with a new pseudonym or new polynomial function due to RA_i's reply. Otherwise, she will receive a registration success reply, she is permitted to use the pseudonym *pse* and polynomial function $f()$ in future transactions.

3.5 Account Guarantee (AGua)

Before executing anonymous identity guarantee phase, the guarantor must be recognized, however, that if the number of accounts the guarantor has insured is equal to or more than the utmost power of the polynomial function, the guarantor faces the possibility of polynomial leaking. The greater the number of accounts guaranteed, the greater the dangers.

To guarantee an account with address *add*, the guarantor must first choose a random number $salt \in Z_n$, and then she generates a guaranty key (xgk, ygk) for the anonymous account as follows.

$$xgk = H(pse, f(), add, salt) \tag{11}$$

$$ygk = f(xgk) \tag{12}$$

Subsequently, the guarantor transmits the following information *acguareq* through a secure connection to RA_i as an anonymous account guarantee request *acguareq*.

$$acguareq := (pse, add, xgk, ygk) \tag{13}$$

When the RA_i accepts the request *acguareq*, she will verify it with the following steps. RA_i obtains the polynomial function $f()'$ from the blockchain with the same pseudonym *pse*. Then she calculates xgk and ygk in the same way for each conceivable salt value (from 0 to n) to see if there would be any salt value that satisfies the result in *acguareq*. In the event that the request is legitimate, the RA_i will sign the guarantee as follows.

1. The RA_i randomly selects $k_g \in Z_p^*$ and calculates auxiliary values as follows.

$$A_3 = g_1^{k_g} \tag{14}$$

$$A_4 = R_i^{x_i} \cdot T^{k_g} \tag{15}$$

$$Q_2 = e(T, S)^{k_g} \tag{16}$$

2. The RA_i calculates the challenge value C_{gu}.

$$m_g = H(add, xgk, ygk) \tag{17}$$

$$C_{gu} = H(m_g, A_3, A_4, Q_2) \tag{18}$$

3. The RA_i sets the address add's witness $W_i = Acc_{old}$ and issues new accumulator's value Acc_{new} as follows.

$$Acc_{new} = Acc_{old}^{l+H(add)} \tag{19}$$

4. Then the RA_i calculates $w_g = k_g * C_{gu} + x_i$. She outputs the certificate $Cert$ as follows.

$$Cert := (pse, add, xgk, ygk, A_3, A_4, C_{au}, w_g, W_i, Acc_{new}) \tag{20}$$

Besides, it broadcasts a join message m_{join} as follows.

$$m_{join} := (state = join, value = H(add), Acc_{join} = Acc_{old}) \tag{21}$$

When the guarantor receives the certificate $Cert$ from the RA_i through a secure connection, she verifies it as the following steps.

1. Firstly, she determines if the account has been joined.

$$e(g_3, Acc_{new}) = e(L \cdot g_3^{H(add)}, W_i) \tag{22}$$

If the equation is hold, the address ass is valid and further verification is performed. Otherwise, the certificate $Cert$ is invalid.

2. Secondly, she should calculate the auxiliary value $\tilde{Q}_2$ as follows.

$$\tilde{Q}_2 = \frac{e\left(A_4, S\right) \cdot e(g_1, g_2)^{w_g}}{e(A_3^{C_{gu}} \cdot D, g_2)} \tag{23}$$

3. The guarantor calculates the challenge value C'_{gu} as follows.

$$m_g = H(add, xgk, ygk) \tag{24}$$

$$C'_{gu} = H(m_g, A_3, A_4, \tilde{Q}_2) \tag{25}$$

4. Finally, the guarantor checks whether $C'_{gu} = C_{gu}$ holds. If true, the guarantor will accept $Cert$ and send it to the owner of the anonymous account. Otherwise, she will reject it.

When other users obtain the message m_{join}, they first check the state to make sure that it is a registration message. Then they update their own membership witness according to the $value$ and Acc_{join}. They compute their new witness W_{new} as follows.

$$W_{new} = W_i^{(value - H(add))} \cdot Acc_{join} \tag{26}$$

3.6 Show Certificate (SCert)

If the user would like to seek the service of the service provider anonymously, she must provide the guarantee $ACert$ she received from the guarantor, which is shown as follows.

$$ACert := (add, xgk, ygk, A_3, A_4, C_{au}, w_g, W_i) \tag{27}$$

3.7 Verify Certificate(VCert)

When the service provider receives the guarantee $ACert$, she will verify its validity. If the verification is passed, she will provide the service to the user and record the corresponding records on the chain, otherwise, she will refuse to provide the service. The specific process of verification is as follows.

1. First, she determines if the account has been revoked. Then she does the following step.

$$e(g_3, Acc_{new}) = e(L \cdot g_3^{H(add)}, W_i) \tag{28}$$

   If the equation holds, the address add is valid and further verification is performed. Otherwise, the authentication fails.
2. The guarantor calculates the challenge value C'_{gu} as follows.

$$m_g = H(add, xgk, ygk) \tag{29}$$

$$C'_{gu} = H(m_g, A_3, A_4, \tilde{Q}_2) \tag{30}$$

3. Finally, the service provider checks whether $C'_{gu} = C_{gu}$ holds.

If true, the service provider accepts $ACert$, providing services and uploading relevant information to the blockchain. Otherwise, she rejects it.

3.8 Supervision (Sup)

When there is a dispute between the service provider and the anonymous user, the service provider provides CA with the block hash of the anonymous certificate presented by the anonymous user. When the CA gets the certificate $ACert$, it will do the following steps.

1. Firstly, it must check the validity of the certificate $ACert$. If the certificate is invalid, then it aborts and outputs $\perp$. Otherwise, it continues.
2. It needs to revoke the user's right to get services, it computes Acc_{new} as follows and issues it.

$$Acc_{new} = Acc_{old}^{1/(H(add)+l)} \tag{31}$$

   She broadcasts a remove message m_{leave}.

$$m_{leave} := (state = revoke, value = H(add), Acc_{leave} = Acc_{new}) \tag{32}$$

3. The CA has a list of polynomial functions $PFList$, it needs to find the suspect polynomial functions according to xgk and ygk, listing a polynomial functions list Sus_PFList.

$$0, 1 \longleftarrow ygk = f(xgk), f() \in PFList \tag{33}$$

4. Then, based on Sus_PFList, CA checks whether the account guarantee keys are generated from a specific user as follows.

$$\begin{aligned} 0, 1 &\longleftarrow xgk = H(pse, f(), add, salt) \\ 0 \leqslant salt &\leqslant n, (pse, f()) \in Sus\_PFList \end{aligned} \tag{34}$$

If a hash result matches xgk, the specific user is a cheater. CA will broadcast it to prevent others from being cheated again.

When others obtain the message m_{leave}, they first check the state to make sure that it is revocation message. Then they update their own membership witness W_{new} according to the $value$ and Acc_{leave} as follows.

$$W_{new} = (W_i / Acc_{leave})^{1/(value - H(add))} \tag{35}$$

4 Analysis

4.1 Security Analysis

Theorem 1. *Without the tracking key and the information of the guarantors, no one can obtain the true identity of the guarantor.*

Proof. In the ERAAS system, when users present their certificates, they need to present the guarantee values xgk and ygk generated by the guarantors and also need to present the signature generated by RA, indicating that their certificate is authorized by RA. At the same time, users also need to present their witness value W_i to prove that they are legitimate and not revoked users. According to [5], when the number of guarantors does not exceed the highest degree of polynomial function, the attacker cannot obtain the polynomial of the guarantor, i.e., the attacker cannot obtain the information of the guarantor through the guarantee values xgk and ygk, which realizes the anonymity of the guarantor.

RA uses the group signature technology in [8] for signing users' certificates. According to [8], when the CA is trusted, i.e., when the tracking key is not leaked, solving for the value A_4/A_3^t is infeasible, which means that RA cannot be found through the anonymous certificates, ensuring the anonymity of RA and guarantors. The dynamic accumulator [18] used in the ERAAS system does not leak any information about the anonymous account or any information related to the guarantor, ensuring the anonymity of the guarantor. Therefore, the ERAAS system supports the anonymity of guarantors.

Theorem 2. *By using the tracing key and the stored information of the guarantors, the real identity of the corresponding guarantor can be traced.*

Proof. In the ERAAS system, when a user presents an anonymous certificate, they need to provide the guarantee values xgk and ygk, and the user needs to show the signature generated by RA for the guarantee to prove that their guarantee is authorized and valid. The user also needs to provide their witness value W_i to prove that their anonymous account is valid. According to [5], if the number of accounts guaranteed by a guarantor does not exceed the highest degree of a polynomial, the adversary cannot recover the polynomial of the guarantor, ensuring the non-forgeability of the guarantor's polynomial.

RA uses group signature technology in [8] to sign the user's guarantee, and according to Theorem 3 in [8], if the ECDL hypothesis holds in a bilinear group, the group signature is traceable. According to [18], the ERAAS system can ensure that the adversary cannot forge the witness value of a revoked user and pass the verification, nor can they forge the witness value of an unauthorized user and pass the verification.

In summary, CA can trace RAs through anonymous certificates. With the guarantee values xgk and ygk on the anonymous certificate and the information of the guarantors stored by the RA, the unique guarantee polynomial can be identified, and therefore, the CA can trace the real identity of the guarantor. Therefore, the ERAAS system satisfies the traceability property.

Theorem 3. *Without the private key of the RA, a malicious account address cannot be re-certified after being removed from the accumulator.*

Proof. In the ERAAS system, when users present anonymous certificates, they need to provide their accumulator witness value W_i to the service provider. According to [18], in the absence of the registration authority's private key, if a user's account in the ERAAS system has been removed from the accumulator, the user cannot re-add the account address to the accumulator, and the witness value W_i provided by the user cannot satisfy $e(g_3, Acc_{new}) = e(L \cdot g_3^{H(add)}, W_i)$, making it impossible to complete the authentication. Therefore, once a malicious account address is removed from the accumulator, users will not be able to use that account address for identity authentication again. Thus, the ERAAS system designed in this paper satisfies revocability.

Theorem 4. *Without the private key of the RA and the polynomials, an attacker cannot forge a valid certificate.*

Proof. When a user provides an anonymous certificate in the ERAAS system, they are required to supply the guarantee values xgk and ygk as well as the signature produced by RA for the guarantee to demonstrate that their guarantee is legitimate. To demonstrate the legitimacy of anonymous account, the user must additionally supply their witness value W_i. According to [5], if the number of accounts it guarantees does not exceed the polynomial's highest degree, the adversary cannot retrieve the guarantor's polynomial, ensuring the non-forgeability of the guarantor's polynomial. Similarly, when issuing a certificate, RA's key is used to sign the certificate, and it is known from [8] that an adversary cannot forge a valid signature. It is also known from [18] that if the adversary

does not have the key, they cannot add an account address to the accumulator and cannot prove the validity of an anonymous account. In summary, without the private key of the RA and the polynomials, adversaries cannot forge polynomials, nor can they forge the RA's signature as the signature of an anonymous certificate, and they cannot add a malicious account address to the accumulator. Therefore, the system proposed in this paper can resist forgery attacks.

4.2 Theoretical Analysis

This section will compare the Identity Mixer scheme [3], the supervised anonymous authentication scheme [23], the anonymous account guarantee scheme [5], and the ERAAS system in terms of functionality. Table 2 summarizes the similarities and differences in functionality between the Identity Mixer scheme [3], the supervised anonymous authentication scheme [23], the anonymous account guarantee scheme [5], and the ERAAS system. In terms of functionality, all of these schemes achieve anonymous authentication. However, the Identity Mixer scheme [3] is completely anonymous, so in case of disputes, it is impossible to trace the person who presented the anonymous certificate. To address this issue, the supervised anonymous authentication scheme [23], the anonymous account guarantee scheme [5], and the ERAAS system introduce regulators to achieve the ability to track the real identity of the malicious users. In order to reduce the storage overhead of certificates, the anonymous account guarantee scheme [5] and the ERAAS system provide guarantee functions, which means that users who only need to be authenticated can find a guarantor to guarantee their anonymous accounts instead of registering with the organization using their real identities. At the same time, the system proposes a revocability function to prevent malicious accounts from performing multiple malicious activities.

Table 2. Functionality comparison

Scheme	Authentication	Anonymous	Supervision	Guaranteeable	Revocable
Camenisch et al. [3]	✓	✓	✗	✗	✗
Wang et al. [23]	✓	✓	✓	✗	✗
Cheng et al. [5]	✓	✓	✓	✓	✗
ERAAS	✓	✓	✓	✓	✓

4.3 Experimental Analysis

We evaluate the performance of ERAAS through experimental analysis. We use the jpbc library and commons-codec-1.7 library in Java to simulate the system and the anonymous account guarantee scheme [5] on a platform with Microsoft Windows 10 operating system, Intel(R) Core(TM) i5-7700 @3.6GHz. The elliptic curve used is of type A ($y^2 = x^3 + x$), with a prime number q of 171 bits.

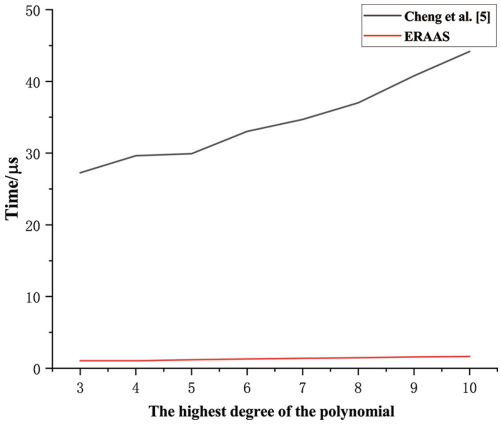

Fig. 2. Comparison of average concurrent processing times in GReg.

During the registration of a guarantor, the RA randomly generates a unique polynomial for each guarantor. However, the time required to randomly generate a polynomial is related to the number of coefficients in the polynomial. According to the data in Fig. 2, we can see that in the GReg phase, the higher the degree of the polynomial, the more coefficients the center needs to generate randomly, so the time to generate the guarantor polynomial will also increase. When fifty guarantors simultaneously send registration requests, Cheng et al.'s scheme [5] only has one center, so only one request can be processed at a time, and other guarantors have to wait, resulting in a high average processing delay. In contrast, the ERAAS system can have multiple RAs processing requests, which can effectively reduce the processing delay for concurrent requests. Therefore, when the highest degree of the polynomial is equal, the processing delay for concurrent registration requests in Cheng et al.'s scheme [5] is greater than that in the ERAAS system. By setting up multiple RAs properly, the system can effectively reduce the registration delay for guarantors.

In the ERAAS system, the AGua phase mainly consists of three stages, including the time for guarantee computation by the guarantors, the time for the RA to sign the guarantee value, and the time for the user to verify the guarantee signature. Since the signature and verification times are fixed, the overall time of the AGua phase mainly depends on the time taken by the guarantors to compute the guarantee value. The computation time of the guarantee value increases with the increase in the highest degree and the maximum coefficient of the polynomial. As shown in Fig. 3, when the highest degree of the polynomial is equal, the larger the maximum coefficient of the polynomial, the longer the time required for AGua. With the same maximum coefficient in the polynomial, higher polynomial degrees result in increased AGua phase time. To reduce AGua phase overhead and enhance guarantee efficiency in ERAAS, lower maximum values for polynomial coefficients and degree should be chosen.

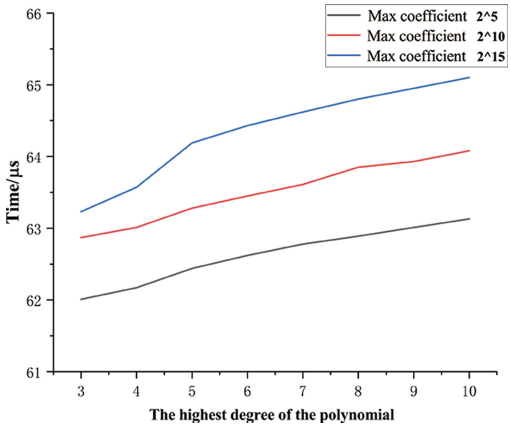

Fig. 3. Processing time in AGua of the ERAAS system.

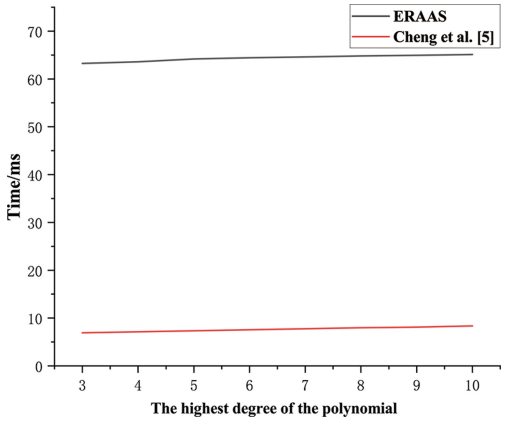

Fig. 4. Processing time in AGua for a single request.

In the AGua phase, if only considering the case of processing a single guarantee request, the efficiency comparison between Cheng et al.'s scheme [5] and the ERAAS system is shown in Fig. 4. As shown in Fig. 4, the time cost of the guarantee increases with the highest degree of the polynomial. When the highest degree of the polynomial generated by the guarantor is the same, Cheng et al.'s scheme [5] has a higher efficiency in processing guarantees than the ERAAS system. This is because the ERAAS system adopts a group signature scheme for guarantee signature generation, and the center needs to add the guaranteed account to the accumulator. While Cheng et al.'s scheme [5] adopts a more efficient ECDSA signature scheme, in the case of processing a single guarantee request, the guarantee efficiency of Cheng et al.'s scheme [5] is higher.

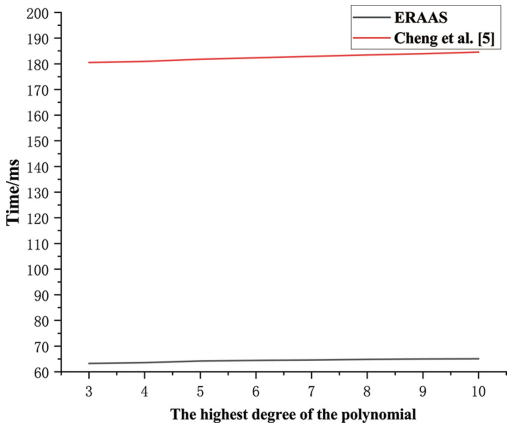

Fig. 5. The average AGua time for concurrent requests.

When the center needs to process fifty concurrent guarantee requests in the AGua phase, the average latency in Cheng et al.'s scheme [5] is longer because the scheme only has a single registration center that can handle one request at a time. In contrast, the ERAAS system has multiple RAs to handle guarantee requests, resulting in shorter guarantee latency, as shown in Fig. 5. Furthermore, from Fig. 5, it can be observed that the guaranteed latency increases with the increase in the highest degree of the polynomial for the same scheme. Therefore, reasonable settings of the RA can improve the system's ability to process concurrent requests and reduce the guaranteed latency. Similarly, choosing smaller values for the maximum coefficient and degree of the polynomial in the ERAAS system can effectively reduce the guaranteed latency.

5 Conclusion

This paper proposed an ERAAS system where blockchain was employed to ensure the non-repudiation of evidence. In the ERAAS system, a user who only needs identity authentication can request a guarantor to provide an anonymous account guaranty for her without the need to apply for a certificate with her real identity, thus reducing the storage cost of certificates. At the same time, the system sets up multiple RAs, and in the case of concurrency, the average latency for users to obtain a guaranty is reduced. In order to prevent malicious accounts from repeating malicious behavior, the ERAAS system also provides a revocation mechanism. Accounts that have engaged in malicious behavior will be revoked and cannot be authenticated again. The security analysis indicated that the proposed ERAAS system satisfies anonymity, traceability and revocability and can resist forgery attacks. The comparison with existing solutions demonstrated that the proposed ERAAS system improves the efficiency of the RA in processing concurrent requests and has good practicality.

Acknowledgements. This article is supported in part by the National Key R&D Program of China under project 2020YFB1006003, the Guangxi Natural Science Foundation under grant 2023GXNSFAA026236, the National Natural Science Foundation of China under projects 62162017, 62172119 and 61962012, the Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ23F020012, the Guangdong Key R&D Program under project 2020B0101090002, and the special fund of the High-level Innovation Team and Outstanding Scholar Program for universities of Guangxi.

References

1. Arasan, A., Sadaiyandi, R., Al-Turjman, F., Rajasekaran, A.S., Selvi Karuppuswamy, K.: Computationally efficient and secure anonymous authentication scheme for cloud users. Personal Ubiquit. Comput. 1–11 (2021). https://doi.org/10.1007/s00779-021-01566-9
2. Banerjee, S., Odelu, V., Das, A.K., Chattopadhyay, S., Park, Y.: An efficient, anonymous and robust authentication scheme for smart home environments. Sensors **20**(4), 1215 (2020)
3. Camenisch, J., et al.: Specification of the identity mixer cryptographic library. IBM Research-Zurich, pp. 1–48 (2010)
4. Cao, Y.N., Wang, Y., Ding, Y., Guo, Z., Wu, Q., Liang, H.: Blockchain-empowered security and privacy protection technologies for smart grid. Comput. Stand. Interfaces **85**, 103708 (2022)
5. Cheng, L., Liu, J., Jin, Y., Li, Y., Wang, W.: Account guarantee scheme: making anonymous accounts supervised in blockchain. ACM Trans. Internet Technol. (TOIT) **21**(1), 1–19 (2021)
6. Gao, T., Deng, X., Guo, N., Wang, X.: An anonymous authentication scheme based on pmipv6 for VANETs. IEEE Access **6**, 14686–14698 (2018)
7. Han, M., Liu, S., Ma, S., Wan, A.: Anonymous-authentication scheme based on fog computing for VANET. PLoS ONE **15**(2), e0228319 (2020)
8. Ho, T.H., Yen, L.H., Tseng, C.C.: Simple-yet-efficient construction and revocation of group signatures. Int. J. Found. Comput. Sci. **26**(5), 611–624 (2015)
9. I'Anson, C., Mitchell, C.: Security defects in CCITT recommendation x. 509: the directory authentication framework. ACM SIGCOMM Comput. Commun. Rev. **20**(2), 30–34 (1990)
10. Jegadeesan, S., Azees, M., Babu, N.R., Subramaniam, U., Almakhles, J.D.: EPAW: efficient privacy preserving anonymous mutual authentication scheme for wireless body area networks (WBANS). IEEE Access **8**, 48576–48586 (2020)
11. Jiang, Y., Ge, S., Shen, X.: AAAS: an anonymous authentication scheme based on group signature in VANETs. IEEE Access **8**, 98986–98998 (2020)
12. Khan, N., Zhang, J., Jan, S.U.: A robust and privacy-preserving anonymous user authentication scheme for public cloud server. Secur. Commun. Netw. 2022 (2022)
13. Lal, N.A., Prasad, S., Farik, M.: A review of authentication methods. Int. J. Sci. Technol. Res. **5**, 246–249 (2016)
14. Liang, W., Wang, Y., Ding, Y., Zheng, H., Liang, H., Wang, H.: An efficient anonymous authentication and supervision system based on blockchain. In: 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC), pp. 306–313. IEEE (2022)
15. Liang, W., Wang, Y., Ding, Y., Zheng, H., Liang, H., Wang, H.: An efficient blockchain-based anonymous authentication and supervision system. Peer-to-Peer Networking and Applications, pp. 1–20 (2023)

16. Liu, H., Sun, Y., Xu, Y., Xu, R., Wei, Z.: A secure lattice-based anonymous authentication scheme for VANETs. J. Chin. Inst. Eng. **42**(1), 66–73 (2019)
17. Mehmood, A., Natgunanathan, I., Xiang, Y., Poston, H., Zhang, Y.: Anonymous authentication scheme for smart cloud based healthcare applications. IEEE access **6**, 33552–33567 (2018)
18. Nguyen, L.: Accumulators from Bilinear Pairings and Applications. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 275–292. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_19
19. Rana, R., Zaeem, R.N., Barber, K.S.: An assessment of blockchain identity solutions: Minimizing risk and liability of authentication. In: 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 26–33. IEEE (2019)
20. Saleem, T., et al.: ProofChain: an x. 509-compatible blockchain-based PKI framework with decentralized trust. Comput. Netw. **213**, 109069 (2022)
21. Wang, F., Xu, G., Gu, L.: A secure and efficient ECC-based anonymous authentication protocol. Secur. Commun. Netw. 2019 (2019)
22. Wang, X., Yan, Z., Zhang, R., Zhang, P.: Attacks and defenses in user authentication systems: a survey. J. Netw. Comput. Appl. **188**, 103080 (2021)
23. Wang, Z., Fan, J., Cheng, L., An, H.Z., Zheng, H.B., Niu, J.X.: Supervised anonymous authentication scheme. J. Softw. **6**, 1705–1720 (2019)
24. Wen, B., Wang, Y., Ding, Y., Zheng, H., Qin, B., Yang, C.: Security and privacy protection technologies in securing blockchain applications. Inf. Sci. **645**, 119322 (2023)
25. Zhang, L., Li, H., Li, Y., Yu, Y., Au, M.H., Wang, B.: An efficient linkable group signature for payer tracing in anonymous cryptocurrencies. Futur. Gener. Comput. Syst. **101**, 29–38 (2019)
26. Zhang, M., Zhou, J., Zhang, G., Zou, M., Chen, M.: EC-BAAS: elliptic curve-based batch anonymous authentication scheme for internet of vehicles. J. Syst. Architect. **117**, 102161 (2021)
27. Zhang, T., Wang, Y., Ding, Y., Wu, Q., Liang, H., Wang, H.: Multi-party electronic contract signing protocol based on blockchain. IEICE Trans. Inf. Syst. **105**(2), 264–271 (2022)
28. Zimmermann, V., Gerber, N.: The password is dead, long live the password-a laboratory study on user perceptions of authentication schemes. Int. J. Hum Comput. Stud. **133**, 26–44 (2020)
29. Zulfiqar, M., Janjua, M.U., Hassan, M., Ahmad, T., Saleem, T., Stokes, J.W.: Tracking adoption of revocation and cryptographic features in x. 509 certificates. Int. J. Inf. Secur. **21**(3), 653–668 (2022)