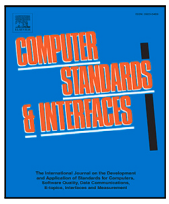
Computer Standards & Interfaces 97 (2026) 104121

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi



Sharing as You Desire: A fuzzy certificateless proxy re-encryption scheme for efficient and privacy-preserving cloud data sharing

Jiasheng Chen [a], Zhenfu Cao [a,*], Liangliang Wang [b,c], Jiachen Shen [a], Xiaolei Dong [a]

[a] East China Normal University, Software Engineering Institute, Shanghai Collaborative Innovation Center of Trusted Industry Internet Software, Shanghai, 200062, China
[b] Shanghai University of Electric Power, Faculty of Artificial Intelligence, Shanghai, 201306, China
[c] Police Integration Computing Key Laboratory of Sichuan Province, Luzhou, 646000, China

ARTICLE INFO

Keywords:
Cloud security
Proxy re-encryption
Certificateless cryptography
Conditional privacy

ABSTRACT

Secure sharing mechanism in the cloud environment not only needs to realize efficient ciphertext storage of resource-constrained clients, but also needs to build a trusted data sharing system. Aiming at the limitations of existing schemes in terms of user identity privacy protection, insufficient access control granularity, and data sharing security, we propose a fuzzy certificateless proxy re-encryption (FCL-PRE) scheme. In order to achieve much better fine-grained delegation and effective conditional privacy, our scheme regards the conditions as an attribute set associated with pseudo-identities, and re-encryption can be performed if and only if the overlap distance of the sender's and receiver's attribute sets meets a specific threshold. Moreover, the FCL-PRE scheme ensures anonymity, preventing the exposure of users' real identities through ciphertexts containing identity information during transmission. In the random oracle model, FCL-PRE not only guarantees confidentiality, anonymity, and collusion resistance but also leverages the fuzziness of re-encryption to provide a certain level of error tolerance in the cloud-sharing architecture. Experimental results indicate that, compared to other existing schemes, FCL-PRE offers up to a 44.6% increase in decryption efficiency while maintaining the lowest overall computational overhead.

1. Introduction

As information technology and the Internet continue to evolve, users can now access networks anytime and anywhere through mobile devices, driving the widespread adoption of cloud services. By leveraging flexible resource scheduling and high network accessibility, cloud computing has attracted enterprises such as Amazon, Google, and Alibaba to introduce cloud-based data storage, access, and sharing services [1–3]. However, cloud service providers are not always completely trustworthy. Due to factors such as technical limitations or economic incentives, they may engage in practices that could compromise users' rights. In recent years, data breaches have occurred frequently: in 2018, Tesla's Kubernetes console on AWS was left unsecured, allowing attackers to exploit the cloud environment; in 2019, Capital One faced misconfigurations on AWS, enabling hackers to gain unauthorized access and disclose more than 100 million user data. Evidently, although outsourcing data to the cloud can reduce the burden of hardware maintenance, it also deprives users of direct control over their data, thereby increasing the risk of potential privacy breaches.

In response to the demand for secure cloud data sharing, the proxy re-encryption (PRE) [4] scheme was proposed. This technology not only allows data to be stored on the cloud server but also capitalizes on the cloud's computing capabilities to securely achieve decryption authorization in Fig. 1. In a typical PRE scheme, key generation center (KGC) is responsible for generating the system's public parameters and issuing public–private key pairs for registered users based on the master secret key. Generally, the data sender encrypts information with their own ID (i.e., e-mail account, phone numbers) and produces the re-encryption key for authorized users, which is stored on the cloud server alongside the ciphertext. Only the authorized recipient can instruct the cloud server to perform ciphertext transformation using the re-encryption key, thereby achieving secure data sharing. However, despite simplifying certificate management, traditional identity-based proxy re-encryption (IB-PRE [5]) still suffers from several limitations: (1) it relies on the KGC for key escrow, meaning that if the KGC is compromised or acts maliciously, users' private keys are at serious risk of exposure; (2) it lacks flexible dynamic authorization, such that even

* Corresponding author.
E-mail addresses: jschen@stu.ecnu.edu.cn (J. Chen), zfcao@sei.ecnu.edu.cn (Z. Cao), llwang@shiep.edu.cn (L. Wang), jcshen@sei.ecnu.edu.cn (J. Shen), dongxiaolei@sei.ecnu.edu.cn (X. Dong).

https://doi.org/10.1016/j.csi.2025.104121
Received 30 June 2025; Received in revised form 23 November 2025; Accepted 21 December 2025
Available online 23 December 2025

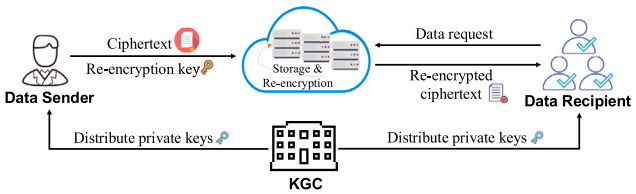


Fig. 1. Data sharing based on proxy re-encryption.

minor changes in a user's identity information require the regeneration of private keys, thus increasing administrative overhead and system complexity; and (3) it struggles to satisfy the requirements of high-privacy scenarios. For instance, in mobile healthcare, patients' private information may be directly used as public keys for encryption [6–8]. Once an attacker traces such identifiers to a patient's real identity, a severe privacy breach can result, endangering the patient's information security.

To address the challenges of insufficient anonymity, key escrow, and difficulty in dynamic privilege adjustment, we propose an anonymous fuzzy certificateless proxy re-encryption scheme (FCL-PRE). Our scheme not only supports identity hiding and fuzzy matching, but also effectively prevents unauthorized access and significantly improves system error tolerance. The main contributions of FCL-PRE are as follows.

- **Fuzzy certificateless PRE with conditional privacy.** A new fuzzy certificateless proxy re-encryption scheme that is tolerant to noisy biometric measurements is proposed. Specifically, the trusted authority first derives a stable, unique biometric identity UID from noisy biometric samples, and then generates a pseudo-identity with a specific set of attributes $\omega = (\omega_i)_{i=1}^{n}$ for it. Re-encryption is allowed only when the overlap between the sender's and receiver's attribute sets satisfies a threshold condition, that is $|\omega \cap \omega'| \geq d$. This policy enforces conditional privacy on top of pseudo-identities, simplifies key management in the certificateless setting, and enables flexible and efficient data sharing among users with similar attributes.
- **Anonymous data sharing via pseudonyms.** The proposed scheme enhances conditional privacy and reduces the cost of managing pseudonyms by tightly binding biometrics, pseudo-identities, and strong keys. The trusted authority internally maintains a mapping $(UID, PUID, \omega)$, where ω is associated with $PUID$. Thus, the privacy-preserving pseudo-identity can only be recovered by the fully trusted authority. Meanwhile, a user can encrypt and share data on behalf of an attribute group using a single $PUID$, rather than maintaining many separate pseudonyms, thus significantly reducing the key management overhead on the user side.
- **Security and practicality.** We provide a detailed security proof of FCL-PRE in the random oracle model, demonstrating that it satisfies chosen plaintext attack (IND-CPA) security. Theoretical analysis and experimental results show that FCL-PRE not only achieves anonymity, error tolerance, and resistance to collusion attack, but also has minimal computational overhead in the decryption phase.

2. Related work

(1) Basic PRE schemes: In 1998, Blaze et al. [4] first introduced the notion of proxy re-encryption (PRE), which enables a semi-honest proxy to transform ciphertexts without accessing the underlying decryption keys. Subsequent early works primarily examined how to delegate decryption capabilities securely and efficiently so as to support data sharing and access control in cloud environments [9–11]. As research progressed, the limitations of the original PRE model gradually became evident. For example, a malicious user may collude with the proxy to recover the sender's private key. Ateniese et al. [12] later presented a unidirectional PRE scheme that offers a certain level of resistance against collusion attacks, although it still depends on a public key infrastructure (PKI) for certificate management. Gentry [13] addressed the burden imposed by PKI by introducing the paradigm of certificate-based cryptography, thereby eliminating the need for online third-party certificate queries. Sur et al. [14] further applied this paradigm by designing a certificate-based encryption scheme. They were the first to combine it with proxy re-encryption, and thus proposed a certificate-based proxy re-encryption (CB-PRE) scheme that achieves chosen-ciphertext (IND-CCA) security in the random oracle model. On the other hand, to further simplify the public key infrastructure, Green and Ateniese [5] extended PRE to identity-based scenarios, significantly reducing certificate management overhead by replacing traditional public keys with user identifiers and achieving adaptive CCA security. In this context, Ge et al. [15] designed an identity-based broadcast PRE (BPRE) scheme that supports revocation of a shared user set and can resist chosen-plaintext attacks, while Zhang et al. [16] employed bilinear pairings to construct an identity-based BPRE scheme for VANETs that achieves CPA security with constant decryption overhead.

(2) Conditional PRE schemes: Once the basic transformation capability of PRE had been established, researchers began to enrich PRE with more expressive access control and privacy guarantees. In traditional PRE systems, once the proxy obtains a re-encryption key, it can often convert all ciphertexts of the delegator for the designated delegatee, which is incompatible with fine-grained authorization requirements. To address this issue, Weng et al. [19] first proposed conditional proxy re-encryption (CPRE). In their construction, a condition expression is embedded into the re-encryption key, so that the proxy is only able to transform ciphertexts that satisfy the specified condition, which enforces strict control over the proxy's capability at the semantic level. At the same time, Ateniese et al. [22] presented a PRE scheme with key privacy. Even if an adversary obtains a re-encryption key, it cannot distinguish the delegatee's identity, which further protects the receiver's privacy. Shao et al. [18] achieved key privacy while preserving CCA security. Li et al. [17] incorporated the idea of conditional PRE into certificate-based cryptography. Their scheme allows only ciphertexts associated with specific subsets to be transformed and forwarded to designated delegatees, and also attains CCA security. In order to support more expressive access structures, Yao et al. [21] designed a CPRE scheme with ciphertext evolution, which ensures that the delegation process remains under the data owner's control. Li et al. [20] proposed a CPRE scheme that supports only a single receiver. Lin et al. [30] developed a CPRE scheme tailored for IoT scenarios, which supports revocation of misbehaving users without relying on a fully trusted third party. Zhang et al. [31] designed a key-sharing mechanism based on CPRE and combined it with a bilinear accumulator to verify the integrity of homomorphic encryption keys stored in the cloud. Chen et al. [25] constructed a conditional BPRE scheme based on bilinear pairings under conditional constraints.

(3) Certificateless-based PRE schemes: Due to the inherent key escrow problem in identity-based cryptography, Sur et al. [32] introduced PRE into the certificateless public key setting [33], and then proposed the concept of certificateless proxy re-encryption (CL-PRE). In CL-PRE, each user's private key is split into a partial private key generated by a key generation center (KGC) and a user-chosen secret value. This design avoids full key escrow by the KGC and does not require traditional certificate management, which makes CL-PRE particularly suitable for resource-constrained environments. Within this framework, Bhatia et al. [34] constructed a lightweight pairing-free CL-PRE scheme and applied it to mobile healthcare scenarios. Eltayieb et al. [35] further adopted blockchain as the proxy to execute the re-encryption

Table 1
Summary of functional comparison with other schemes.

Schemes	Techniques	Conditional privacy	Fuzzy matching	Anonymity	Multiple receivers	Collusion resistance
[13,14,17]	CB-PRE	×	×	×	×	✓
[18]	CPRE	✓	×	✓	✓	×
[15,16]	IB-PRE	×	×	×	✓	✓
[19,20]	CPRE	✓	×	×	×	×
[21]	IB-CPRE	✓	×	×	✓	✓
[22]	CPRE	✓	×	✓	×	×
[23,24]	CL-PRE	×	×	×	✓	✓
[25]	IB-CPRE	✓	×	✓	✓	✓
[26,27]	Fuzzy IB-CPRE	✓	✓	×	✓	×
[28,29]	CL-CPRE	✓	×	×	✓	✓
Ours	Fuzzy CL-CPRE	✓	✓	✓	✓	✓

algorithm, which not only preserves data confidentiality but also provides a flexible revocation mechanism. Subsequent CL-PRE works [23,24,36] mainly focused on improving efficiency, supporting revocation, and enhancing traceability. Similarly, to prevent cloud platforms from abusing re-encryption permissions, Li et al. [28] proposed a novel pairing-free scheme based on certificateless conditional BPRE. Zhou et al. [29] combined certificateless public key cryptography and PRE, which realizes multi-level data access control, dynamic key update, and ciphertext evolution.

(4) Fuzzy PRE schemes: In another line of research, advances in biometric technologies have introduced new design dimensions for PRE. Fuzzy identity-based encryption (FIBE) [37] leverages biometric characteristics such as fingerprints and irises, which are inherently unique and tamper-resistant, to derive descriptive attribute sets that serve as a natural attribute space for encryption and authorization. Following this idea, Fang et al. [26] proposed an FCPRE scheme in which descriptive keywords are used as conditions to realize fuzzy conditional PRE. In their scheme, the proxy can re-encrypt ciphertexts according to a t-out-of-d threshold strategy. Xiong et al. [38] later proposed an improved pairing-based fuzzy identity-based signature (FIBS) scheme that supports the error tolerance property. Li et al. [27] presented the first lattice-based FIB-CPRE scheme. Their scheme provides finer-grained control over delegated decryption, but incurs high computational cost, which negatively affects overall encryption and decryption efficiency. It should be noted that the use of biometric traits can significantly improve usability, but the noise inevitably introduced during biometric acquisition and feature extraction makes key generation and matching more challenging. To cope with this issue, Wang et al. [39] proposed a novel fuzzy certificateless signature authentication scheme that achieves conditional privacy while effectively protecting the confidentiality of users' real biometric characteristics.

As summarized in Table 1, existing PRE schemes and their variants have achieved substantial progress in terms of functionality and applicability to diverse scenarios. However, several important limitations remain.

- The scalability on the receiver side is restricted. Many schemes do not efficiently support data sharing among multiple receivers, which limits their practicality in large-scale collaborative applications, such as schemes [14,17,20].
- The strong binding between real identities and biometric characteristics introduces significant privacy risks. Some biometric-based schemes do not adequately protect the identity privacy of senders and receivers, and therefore cannot satisfy stringent privacy requirements, as in schemes [23,24,26,28,29].

3. Preliminaries

This section briefly overviews the basic concepts and techniques discussed in our scheme. Table 2 provides a list of symbols and their descriptions.

3.1. Bilinear map

Suppose there exists a mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ represent two cyclic groups with the same prime order q. P is a generator of $\mathbb{G}$, then a bilinear map e should have the following properties [40]:

- Bilinearity: $e(aP, bP) = e(P, P)^{ab}$ holds for all $a, b \in Z_q^*$.
- Nondegeneracy: There exists P such that $e(P, P) \neq 1$.
- Computability: $e(P_1, P_2)$ can be computed efficiently for all $P_1, P_2 \in \mathbb{G}$.

3.2. Useful definitions

Definition 1 (*Shamir Secret Sharing [41]*)**.** Shamir's secret sharing scheme, introduced in 1979, is based on polynomial interpolation. A secret s is divided into n shares, denoted as $s_1, \ldots, s_n$ with a threshold t, such that any set of at least t participants $\mathcal{P}_i$ can recover s, whereas any subset of size less than t gains no information about it. The scheme consists of the following phases:

- Secret distribution: Let $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ denote the set of participants and randomly select the secret value $s \in Z_q^*$. Then, a polynomial $F(x)$ of degree $t-1$ is selected that satisfying the condition of $F(0) = s$, then $F(x)$ can be expressed as:

  $$F(x) = s + \sum_{j=1}^{t-1} a_j x^j \bmod q.$$

  Therefore, the share set $SS = \{(\omega_i, s_i) | 1 \leq i \leq n\}$, where $F(\omega_i) = s_i$. The ith share (ω_i, s_i) is privately delivered to the corresponding participant $\mathcal{P}_i$.
- Secret reconstruction: Let $S \subseteq \{1, \ldots, n\}$ be a group with $|S| = t$. The secret value is reconstructed from shares $s_1, \ldots, s_n$ using the Lagrange interpolation method:

  $$F(x) = \sum_{\mathcal{P}_i \in S} \Delta_{\omega_i, S}(x) F(\omega_i) = \sum_{\mathcal{P}_i \in S} \Delta_{\omega_i, S}(x) s_i.$$

  where $\Delta_{\omega_i, S}(x) = \prod_{\mathcal{P}_i \in S, k \neq i} \frac{x - \omega_k}{\omega_i - \omega_k}$ is denoted as the Lagrange coefficient.

Definition 2 (*Decisional Bilinear Diffie–Hellman (DBDH) Assumption*)**.** Given a random instance (P, aP, bP, cP, T), $P \in \mathbb{G}$, a, b, c are randomly selected elements from Z_q^*, and T is an element in $\mathbb{G}_T$. The DBDH assumption requires determining whether T is equal to $e(P, P)^{abc}$ or a random element in $\mathbb{G}_T$. For any PPT algorithms $\mathcal{A}$, the advantage of successfully distinguishing between $T = e(P, P)^{abc}$ and a random element is defined as follows.

$$Adv_{\mathcal{A}}^{DBDH}(\lambda) = |Pr[\mathcal{A}(P, aP, bP, cP, e(P, P)^{abc}) = 1]| - |Pr[\mathcal{A}(P, aP, bP, cP, T) = 1]|$$

If the advantage $Adv_{\mathcal{A}}^{DBDH}(\lambda)$ in solving the DBDH is negligible, then the DBDH assumption holds.

Table 2
Summary of notations.

Symbol	Description
λ	Security parameter
msk	Master secret key
bio	Biometric characteristic
$IdGen(\cdot)$	An identity extraction function
UID	Realistic identity
$PUID$	Pseudo-identity
d	Error tolerance
ω	An attribute set
x_{PUID}	Secret value
SK_{PUID}	User's full private key
PK_{PUID}	Public key
$RK_{S,\omega,\mathcal{R}}$	Re-encryption key
CT	Original ciphertext
CT'	Re-encrypted ciphertext

Definition 3 (*Syntax of FCL-PRE*)**.** The nine polynomial-time algorithms shown below constitute our FCL-PRE scheme.

- Setup. On input a security parameter λ, TA and KGC generate system parameter $params$, and a master secret key msk that is kept secret from user.
- PartialPrivateKey. After TA publishes the pseudo-identity $PUID$ for each registered user, KGC generates the corresponding partial private key D_{PUID} and sends it to the user.
- SetSecretValue. The sender $\mathcal{S}$ executes the algorithm, and chooses a secret value x_{PUID} randomly.
- SetPrivateKey. On input $PUID$, $params$, x_{PUID} and D_{PUID}, $\mathcal{S}$ generates the complete private key SK_{PUID}.
- SetPublicKey. $\mathcal{S}$ performs this algorithm, and inputs x_{PUID}, then outputs the full public key PK_{PUID}.
- Encryption. On input $PUID$, $params$, a message m, and PK_{PUID}, $\mathcal{S}$ computes the original ciphertext CT.
- ReKey Generation. Given the private key SK_{PUID}, $\mathcal{R}$'s pseudo-identity $PUID'$ and the corresponding $PK_{PUID'}$, $\mathcal{S}$ generates a conditional re-encryption key $RK_{S,\omega,\mathcal{R}}$ by running this algorithm.
- Re-encryption. Upon receiving $RK_{S,\omega,\mathcal{R}}$, the original ciphertext CT, the cloud should verify whether the equation $|\omega \cap \omega'| \geq d$ holds. If and only when the algorithm satisfies, the original ciphertext CT can be re-encrypted, and the second-layer of ciphertext CT' can be generated.
- Decryption. The user invokes it to decrypt the corresponding ciphertext, resulting in either the plaintext m or $\perp$.

4. Scheme model

In this section, we introduce the system model, outline the security guarantee model, and specify security requirements, respectively.

4.1. System model

The operation flow of fuzzy certificateless proxy re-encryption scheme is shown in Fig. 2. It includes five different parties, namely: Trusted Authority, Key Generation Center, Cloud Proxy Server, Sender, and Receiver.

- **Trusted Authority (TA):** TA is a fully trusted authority whose primary role is to generate privacy-preserving pseudo-identities $PUID$ for users and to cooperate with KGC in setting up and publishing the public parameters. At the same time, it maintains an internal mapping $(UID, PUID, \omega)$, where ω denotes the attribute set associated with each $PUID$. Only the pseudo-identity and its associated attribute information are exposed to other entities, while the real identity UID remains exclusively known to TA.

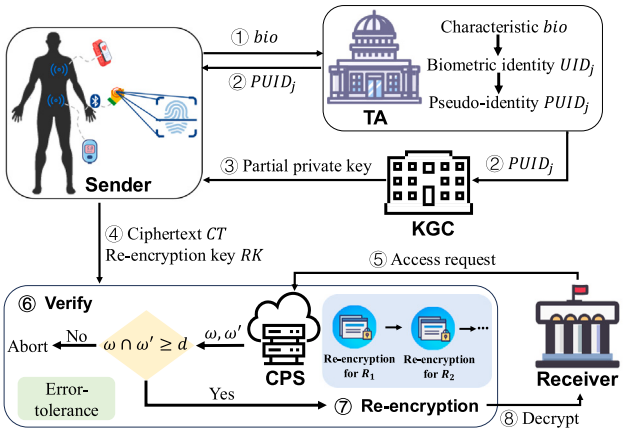


Fig. 2. The operation flow of FCL-PRE.

- **Key Generation Center (KGC):** As an honest but curious KGC, it is responsible for performing system initialization and generating a partial private key related to the user's identity, and it is assumed that KGC and TA will not collude.
- **Cloud Proxy Server (CPS):** CPS is responsible for storing original ciphertexts and executing conditional re-encryption operations. When the receiver $\mathcal{R}$ sends an access request, CPS first verifies whether the condition $|\omega \cap \omega'| \geq d$. If so, sender $\mathcal{S}$ generates a corresponding re-encryption key for CPS to perform re-encryption. Otherwise, CPS refuses to implement the re-encryption operation. Please note that, as a semi-trusted entity, it may still attempt to infer user privacy from the shared data.
- **Sender ($\mathcal{S}$):** $\mathcal{S}$ can use the public key associated with $PUID$ to encrypt the data to be shared, generate the original ciphertext CT and upload it to CPS storage. In addition, $\mathcal{S}$ produces the corresponding re-encryption key $RK_{S,\omega,\mathcal{R}}$ according to the result of the verification equation, and sends it to CPS.
- **Receiver ($\mathcal{R}$):** The authorized receiver $\mathcal{R}$ can decrypt and obtain the plaintext by downloading the re-encrypted ciphertext.

4.2. Security guarantee model

There are two types of adversaries in the certificateless cryptosystem [42]: $\mathcal{A}_1$ is the first type of adversary, which can replace the user's public key, and $\mathcal{A}_2$ is the second type of adversary, which can obtain the master secret key. **Game-I** and **Game-II** are the IND-CPA security games for FCL-PRE. Please note that each pseudo-identity $PUID$ is associated with an attribute set ω.

Game-I. This game embodies the attack ability of $\mathcal{A}_1$, challenger $\mathcal{B}$ responds to $\mathcal{A}_1$'s a series queries by controlling the following oracles.

- **Initialization.** When λ is received, $\mathcal{B}$ first executes the Setup algorithm to obtain $params$, and generates the system master key msk. Then, $\mathcal{B}$ outputs $params$ and keeps msk in secret.
- **Phase 1.** The adversary $\mathcal{A}_1$ initiates a series of queries, and $\mathcal{B}$ responds accordingly.
  - PPKQuery oracle $\mathcal{O}_{ppk}$: $\mathcal{B}$ executes the PartialPrivateKey algorithm to generate the partial private key D_{PUID} for the $PUID$ and returns it to $\mathcal{A}_1$.
  - SKQuery oracle $\mathcal{O}_{sk}$: After receiving the partial private key D_{PUID}, $\mathcal{B}$ first runs PartialPrivateKey and SetSecretValue algorithms to obtain the corresponding D_{PUID} and x_{PUID}. Next, $\mathcal{B}$ runs the SetPrivateKey algorithm to generate the complete private key SK_{PUID}, and returns it to $\mathcal{A}_1$.

- PKQuery oracle $\mathcal{O}_{pk}$: $\mathcal{B}$ runs the SetSecretValue algorithm to obtain x_{PUID}, and extracts the user's public key PK_{PUID} by running the SetPublicKey algorithm. Finally, $\mathcal{B}$ returns it to $\mathcal{A}_1$.
- PK replacement oracle $\mathcal{O}_{pkrp}$: When $\mathcal{A}_1$ queries a two-tuple $(PUID, \widetilde{PK_{PUID}})$, where $\widetilde{PK_{PUID}}$ is the newly selected public key to replace the public key PK_{PUID} currently associated with $PUID$. Therefore, $\mathcal{A}_1$ performs public key replacement, such as $PK_{PUID} = \widetilde{PK_{PUID}}$.
- ReKeyGen oracle $\mathcal{O}_{rk}$: $\mathcal{B}$ runs the ReKey Generation algorithm and returns a re-encryption key $RK_{S,\omega,\mathcal{R}}$ to $\mathcal{A}_1$. If the public key of $PUID$ has been replaced at this time, $\mathcal{A}_1$ cannot perform this query.
- Re-encryption oracle $\mathcal{O}_{reen}$: $\mathcal{B}$ performs it and returns a re-encrypted CT' to $\mathcal{A}_1$. If the public key of $PUID$ has been replaced, $\mathcal{A}_1$ cannot perform the query.

- **Challenge.** After completing all the interactions between $\mathcal{A}_1$ and $\mathcal{B}$, $\mathcal{A}_1$ outputs a challenge identity $PUID_\pi$ and two messages of equal length (m_0, m_1). $\mathcal{B}$ randomly selects a message m_b, $b \in \{0,1\}$, calculates the corresponding ciphertext and returns it to $\mathcal{A}_1$.
- **Phase 2.** $\mathcal{A}_1$ and challenger $\mathcal{B}$ continue to conduct queries and answers similar to phase 1, but must follow three constraints.

  (1) $\mathcal{A}_1$ has never queried the partial private key or private key for the challenge identity $PUID_\pi$ that meets the $|\omega \cap \omega_\pi| \geq d$.
  (2) If $\mathcal{A}_1$ sends the re-encryption key queries to a challenge identity $PUID_\pi$ that meets the $|\omega \cap \omega_\pi| \geq d$ condition, then the partial private key queries or private key queries can no longer be performed.
  (3) If $\mathcal{A}_1$ has sent the partial private key or private key queries to challenge identity $PUID_\pi$ that meets the $|\omega \cap \omega_\pi| \geq d$ condition, the re-encryption key queries can no longer be performed, and the information related to the re-encrypted ciphertext cannot be queried.

- **Guess.** Finally, $\mathcal{A}_1$ guesses the challenge bit $b' \in \{0,1\}$. If $b' = b$, $\mathcal{A}_1$ wins this game.

Definition 4. According to the definition of Game-I, our FCL-PRE is IND-CPA secure if the advantage of $\mathcal{A}_1$ is negligible, defined as

$$Adv_{\mathcal{A}_1}^{Game-I}(\lambda) = |Pr[b' = b] - \frac{1}{2}|.$$

Game-II. The game embodies the attack ability of $\mathcal{A}_2$, challenger $\mathcal{B}$ responds to $\mathcal{A}_2$'s a series queries by controlling the following oracles. Game-II is similar to Game-I, therefore, only their main differences are presented below.

- **Initialization.** When λ is received, $\mathcal{B}$ first executes the Setup algorithm to obtain $params$, and generates a system master key msk. Then, $\mathcal{B}$ returns them to $\mathcal{A}_2$.
- **Phase 1.** $\mathcal{A}_2$ issues a series of queries similar to those in Game-I, and $\mathcal{B}$ responds accordingly. At this time, $\mathcal{A}_2$ lacks the ability to replace the public key.
- **Challenge.** Similar to the Game-I.
- **Phase 2.** $\mathcal{A}_2$ and challenger $\mathcal{B}$ continue to conduct similar queries and answers as in phase 1, but must follow three constraints.

  (1) $\mathcal{A}_2$ has never queried the private key for the challenge identity $PUID_\pi$ that meets the $|\omega \cap \omega_\pi| \geq d$ condition.
  (2) If $\mathcal{A}_2$ sends the re-encryption key queries to a challenge identity $PUID_\pi$ that meets the $|\omega \cap \omega_\pi| \geq d$ condition, then the private key queries can no longer be performed.
  (3) If $\mathcal{A}_2$ has sent the private key queries to the challenge identity $PUID_\pi$ that meets the $|\omega \cap \omega_\pi| \geq d$ condition, the re-encryption key queries can no longer be performed, and the information related to the re-encrypted ciphertext cannot be queried.

- **Guess.** Finally, $\mathcal{A}_2$ guesses the challenge bit $b' \in \{0,1\}$. If $b' = b$, $\mathcal{A}_2$ wins this game.

Definition 5. According to the definition of Game-II, our FCL-PRE is IND-CPA secure if the advantage of $\mathcal{A}_2$ is negligible, defined as

$$Adv_{\mathcal{A}_2}^{Game-II}(\lambda) = |Pr[b' = b] - \frac{1}{2}|.$$

4.3. Security requirements

The proposed FCL-PRE scheme should satisfy the following security objectives.

- *Confidentiality.* FCL-PRE must protect sensitive information before it is uploaded to the CPS and prevent any access by unauthorized recipients. Additionally, when generating the original ciphertext and re-encryption key, conditional information is incorporated to ensure that re-encryption can only be performed if the original ciphertext meets specific conditions.
- *Anonymity.* To protect user privacy, FCL-PRE must conceal the user's real biometric identity. Unless it is a trusted third party, no adversary can establish a valid biometric identification association, thereby preventing the leakage of the user's identity information.
- *Error tolerance.* Considering that biometric characteristic may contain some noise with each sampling, FCL-PRE must exhibit error tolerance. Specifically, when the distance between the biometric identity ω of the sender S and another identity ω' is higher than a predefined threshold d, the proxy can use the re-encryption key to generate the corresponding re-encrypted ciphertext for ω', enabling efficient data sharing.
- *Collusion resistance.* In our FCL-PRE, even in the presence of semi-trusted parties, such as collusion between CPS and the receiver, CPS cannot obtain the sender's complete private key and thus cannot perform any decryption operations, ensuring the system's security against internal collusion attacks.

5. The proposed FCL-PRE scheme

In this section, we thoroughly describe FCL-PRE, which supports efficient fuzzy data sharing through anonymized biometric identities. The procedure flow of FCL-PRE is presented in Fig. 3.

5.1. System initialization

(1) Upon inputting the security parameter λ, KGC generates a bilinear pairing parameters $(e, \mathbb{G}, \mathbb{G}_T, q, P)$, where $\mathbb{G}$ and $\mathbb{G}_T$ represent two cyclic groups with the same prime order q, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, P is the generator of $\mathbb{G}$. Then, KGC selects $s \in Z_q^*$ randomly and calculates the system public key $P_{pub} = sP$.
(2) TA considers a symmetric key encryption scheme to hide the user's realistic identity UID, denoted by $Enc_\phi(\cdot)$ and $Dec_\phi(\cdot)$. Here, $Enc_\phi(\cdot)$ represents the encryption algorithm, $Dec_\phi(\cdot)$ represents the decryption algorithm, and ϕ is the shared symmetric key.
(3) Finally, TA and KGC choose four collision-resistant hash functions: $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \{0,1\}^* \to \mathbb{G}$, $H_3 : \{0,1\}^* \to \mathbb{G}$, and $H_4 : \{0,1\}^* \to Z_q^*$, define the system parameters as $params = \{\mathbb{G}, \mathbb{G}_T, e, q, d, P, P_{pub}, H_1, H_2, H_3, H_4\}$.

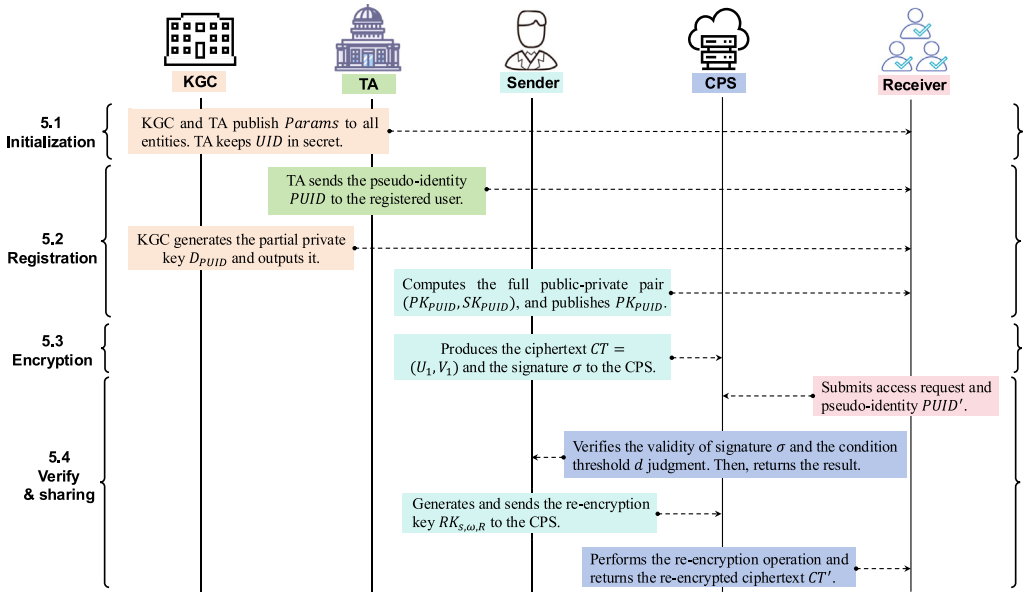


Fig. 3. The algorithm procedure of FCL-PRE.

5.2. User registration phase

Before sharing data, each user must register their identity information with TA. Let the sender be denoted as S_j. First, S_j transmits the realistic biometric information *bio* (i.e., fingerprint) to TA via a secure channel. Then, TA applies the identity extraction function $IdGen(\cdot)$ to convert *bio* into a unique biometric identity $UID_j = IdGen(bio)$. The $IdGen(\cdot)$ function is similar to a hash function and is irreversible. It transforms the biometrics into an identity that is indistinguishable from random information and cannot be used to infer the original biometrics [39,41].

Next, TA generates a pseudo-identity as $PUID_j = Enc_\phi(UID_j \parallel n_{PUID}) \parallel T_j$ to protect the real biometric identity, where n_{PUID} represents the number of pseudo-identities requested and T_j is the validity period of the pseudo-identity. Meanwhile, TA internally maintains a mapping $(UID_j, PUID_j, \omega)$, where ω is the attribute set associated with $PUID_j$. Eventually, TA publishes $PUID_j$ and keeps UID_j secret.

(1) Upon receiving the attribute set ω associated with S_j's pseudo-identity $PUID_j$, KGC first randomly selects a polynomial $p(x)$ of degree $d-1$ such that $p(0) = s$ and assigns $p(\omega_i) = s_i$, where $i \in \{1, \ldots, n\}$. Then it calculates the partial private key as $D_{i,j} = s_i H_1(PUID_j)$. The partial private key $(D_{i,j})_{i=1}^n$ of S_j is represented by KGC as D_{PUID_j}.

(2) After receiving the partial private key D_{PUID_j}, S_j can calculate Lagrange coefficients and perform local verification to ensure consistency: $e(D_{PUID_j}, P) = e(H_1(PUID_j), P_{pub})$. Then, S_j chooses a random secret value $x_{PUID_j} \in Z_q^*$, a polynomial $y(x)$ of degree $d-1$ such that $y(0) = x_{PUID_j}$, and lets $y(\omega_i) = x_{i,PUID_j}$, where $i \in \{1, \ldots, n\}$. Then, S_j's secret value $(x_{i,PUID_j})_{i=1}^n$ is defined as x_{PUID_j}.

(3) Obtaining D_{PUID_j}, S_j sets the full private key as $SK_{PUID_j} = (D_{PUID_j}, x_{PUID_j})$.

(4) S_j calculates $PK_{PUID_j} = x_{PUID_j} P$ as the public key, and publishes it.

5.3. Data encryption phase

Given the S_j's identity $PUID_j$ associated with an attribute set $\omega = (\omega_i)_{i=1}^n$, the public key PK_{PUID_j}, and a message m.

(1) S_j picks a random number $r_j \in Z_q^*$, and a polynomial $g(x)$ of degree $d-1$ such that $g(0) = r_j$ and assigns $g(\omega_i) = r_{i,j}$, where $i \in \{1, \ldots, n\}$. Then, S_j computes

$$U_1 = r_j P, E_j = H_2(PUID_j \parallel PK_{PUID_j} \parallel P_{pub}),$$
$$V_1 = m \prod_{\omega_i \in S} (e(P_{pub}, H_1(PUID_j))^{r_{i,j}} \times e(PK_{PUID_j}, E_j)^{r_{i,j}})^{\Delta_{\omega_i, S}(0)}$$

S_j uploads the original ciphertext $CT = (U_1, V_1)$ to the CPS.

(2) Finally, S_j selects $k \in Z_q^*$ randomly, and computes $R = kP$, $h = H_4(U_1 \parallel V_1 \parallel R \parallel PK_{PUID_j} \parallel PUID_j)$. Then, S_j generates a signature $\sigma_j = k + h \cdot x_{PUID_j} \mod q$, and transmits (R, σ) to the CPS.

5.4. Verification and sharing phase

When a new receiver $\mathcal{R}_j$ initiates an access request, $\mathcal{R}_j$ first needs to send the current pseudo-identity to CPS. After the identity authentication is successful, CPS performs re-encryption operations based on this pseudo-identity.

(1) The CPS first computes $h' = H_4(U_1 \parallel V_1 \parallel R \parallel PK_{PUID'_j} \parallel PUID'_j)$ and $\sigma_j P \stackrel{?}{=} R + h' \cdot PK_{PUID'_j}$. After the signature verification is successful, CPS selects a d-element subset, $S \subseteq \omega \cap \omega'$ randomly, and determines whether the input attribute set ω' satisfies $|\omega \cap \omega'| \geq d$, if yes, CPS returns the result to the sender.

(2) S_j generates the corresponding re-encryption key for the pseudo-identity based on the result. S_j computes $\varphi = e(D_{PUID_j}, H_1(PUID'_j))$, $RK_{S,\omega,\mathcal{R}} = -D_{PUID_j} - x_{PUID_j} E_j + H_3(\varphi \parallel x_{PUID_j} PK_{PUID'_j} \parallel \omega \parallel \omega')$, and then sends $RK_{S,\omega,\mathcal{R}}$ to CPS.

(3) Finally, CPS can use the re-encryption key $RK_{S,\omega,\mathcal{R}}$ to convert CT into a re-encrypted ciphertext CT'. It computes $U_2 = U_1$, $V_2 = V_1 e(U_1, RK_{S,\omega,\mathcal{R}})$, and then outputs $CT' = (U_2, V_2)$ to the authorized recipient.

5.5. Data decryption phase

The procedure to decrypt the original ciphertext and the re-encrypted ciphertext is as follows:

Correctness

For the original ciphertext $CT = (U_1, V_1)$:

$$m = \frac{V_1}{\prod_{\omega_i \in S} e(U_1, D_{PUID_j} + x_{PUID_j} E_j)^{\Delta_{\omega_i, S}(0)}}$$
$$= \frac{m \prod_{\omega_i \in S} (e(P_{pub}, H_1(PUID_j))^{r_{i,j}} \times e(PK_{PUID_j}, E_j)^{r_{i,j}})^{\Delta_{\omega_i, S}(0)}}{\prod_{\omega_i \in S} e(U_1, D_{PUID_j} + x_{PUID_j} E_j)^{\Delta_{\omega_i, S}(0)}}$$
$$= \frac{m}{\prod_{\omega_i \in S} (\frac{e(U_1, D_{PUID_j} + x_{PUID_j} E_j)}{e(P_{pub}, H_1(PUID_j))^{r_{i,j}} \times e(PK_{PUID_j}, E_j)^{r_{i,j}}})^{\Delta_{\omega_i, S}(0)}}$$
$$= \frac{m}{\frac{e(r_j P, \sum_{\omega_i \in S} (p(\omega_i) \Delta_{\omega_i, S}(0)) H_1(PUID_j)) e(r_j P, x_{PUID_j} E_j)}{e(sP, \sum_{\omega_i \in S} (g(\omega_i) \Delta_{\omega_i, S}(0)) H_1(PUID_j)) e(x_{PUID_j} P, \sum_{\omega_i \in S} (g(\omega_i) \Delta_{\omega_i, S}(0)) E_j)}}$$
$$= \frac{m}{\frac{e(r_j P, sH_1(PUID_j)) e(r_j P, x_{PUID_j} E_j)}{e(sP, r_j H_1(PUID_j)) e(x_{PUID_j} P, r_j E_j)}} = m$$

For the re-encrypted ciphertext $CT' = (U_2, V_2)$:

$$m = \frac{V_2}{\prod_{\omega_i \in S} e(U_2, H_3(\varphi \| x_{PUID'_j} PK_{PUID_j} \| \omega \| \omega'))^{\Delta_{\omega_i, S}(0)}}$$
$$= \frac{m \prod_{\omega_i \in S} (e(P_{pub}, H_1(PUID_j))^{r_{i,j}} \times e(PK_{PUID_j}, E_j)^{r_{i,j}})^{\Delta_{\omega_i, S}(0)} e(U_1, RK_{S, \omega, \mathcal{R}})}{\prod_{\omega_i \in S} e(U_2, H_3(\varphi \| x_{PUID'_j} PK_{PUID_j} \| \omega \| \omega'))^{\Delta_{\omega_i, S}(0)}}$$
$$= \frac{me(sP, r_j H_1(PUID_j)) e(x_{PUID_j} P, r_j E_j) e(r_j P, -D_{PUID_j} - x_{PUID_j} E_j + H_3(\varphi \| x_{PUID_j} PK_{PUID'_j} \| \omega \| \omega'))}{e(r_j P, H_3(\varphi \| x_{PUID'_j} PK_{PUID_j} \| \omega \| \omega'))}$$
$$= \frac{me(r_j P, D_{PUID_j} + x_{PUID_j} E_j) e(r_j P, -D_{PUID_j} - x_{PUID_j} E_j) e(r_j P, H_3(\varphi \| x_{PUID_j} PK_{PUID'_j} \| \omega \| \omega')))}{e(r_j P, H_3(\varphi \| x_{PUID'_j} PK_{PUID_j} \| \omega \| \omega'))} = m$$

(1) For the original ciphertext CT, sender S_j can get the plaintext by computing

$$m = \frac{V_1}{\prod_{\omega_i \in S} e(U_1, D_{PUID_j} + x_{PUID_j} E_j)^{\Delta_{\omega_i, S}(0)}}$$

(2) For the re-encrypted ciphertext CT', only authorized receivers can successfully obtain the data.

$$m = \frac{V_2}{\prod_{\omega_i \in S} e(U_2, H_3(\varphi \| x_{PUID'_j} PK_{PUID_j} \| \omega \| \omega'))^{\Delta_{\omega_i, S}(0)}}$$

6. Security analysis

6.1. Security proof for FCL-PRE

Theorem 1. *If adversary $\mathcal{A}_1$ breaks FCL-PRE with a non-negligible advantage ε, we can construct an algorithm $\mathcal{B}$ that solves the DBDH assumption in polynomial time with an advantage ε'.*

Proof. Given a set of challenge instance (P, aP, bP, cP, T), $\mathcal{B}$ acts as a subroutine of the adversary $\mathcal{A}_1$ and attempts to determine whether $T = e(P, P)^{abc}$. Therefore, $\mathcal{B}$ needs to answer a series of inquiries from $\mathcal{A}_1$.

- **Initialization.** By executing Setup algorithm, $\mathcal{B}$ gets $params = \{\mathbb{G}, \mathbb{G}_T, q, e, d, P, P_{pub}, H_1, H_2, H_3\}$. Then, $\mathcal{B}$ sets $P_{pub} = aP$, and a is the master key, which is unknown to $\mathcal{B}$.
  - H_1 Query: $\mathcal{B}$ maintains an initially empty list of the form $L_1(PUID, (h_{1i})_{i=1}^n, (z_i)_{i=1}^n, \alpha_u)$, $\mathcal{A}_1$ publishes $PUID$ for query. $\mathcal{B}$ first chooses $\pi \in \{1, 2, \ldots, q_{H_1}\}$ and defines $PUID_\pi$ as the challenge identity. If $PUID$ already exists in the L_1, $\mathcal{B}$ restores the corresponding record and returns $H_1(PUID) = (h_{1i})_{i=1}^n$ to $\mathcal{A}_1$. Otherwise, for this tuple, $\mathcal{B}$ considers the following two cases:
    - Case 1: If $|\omega \cap \omega_\pi| \geq d$, $\mathcal{B}$ randomly selects a polynomial $t(x)$ of degree $d-1$ such as $t(0) = h$, and returns h to $\mathcal{A}_1$. Then, $\mathcal{B}$ saves the tuple $(PUID, h, \perp, \perp)$ in the L_1.
    - Case 2: If $|\omega \cap \omega_\pi| < d$, $\mathcal{B}$ need to selects $\alpha_u \in \{0, 1\}$ at random, where the probability of $\alpha_u = 1$ is γ.
      (1) When $\alpha_u = 0$, $\mathcal{B}$ chooses a random number $z_i \in Z_q^*$, a polynomial $y(x)$ of degree $d-1$, $y(0) = z$. Let $z_i = y(\omega_i)$, where $i = \{1, \ldots, n\}$, $\mathcal{B}$ calculates $H_1(PUID) = z_i cP$, and saves tuple $(PUID, z_i cP, (z_i)_{i=1}^n, 0)$ in the L_1.
      (2) When $\alpha_u = 1$, $\mathcal{B}$ selects $z^* \in Z_q^*$, outputs $H_1(PUID) = z^* P$ and saves tuple $(PUID, z^* P, z^*, 1)$ in the L_1.
  - H_2 Query: $\mathcal{B}$ maintains an initially empty list of the form $L_2(PUID, t_i, Y_i)$. When $\mathcal{A}_1$ makes a query, if $PUID$ already exists in the L_2, $\mathcal{B}$ answers with Y_i, otherwise it randomly selects $t_i \in Z_q^*$, calculates $Y_i = t_i P$ and adds the tuple $(PUID, t_i, Y_i)$ to the L_2.
  - H_3 Query: $\mathcal{B}$ maintains an initially empty list of the form $L_3(X', H')$. If X' is in the list L_3, $\mathcal{B}$ returns H' to $\mathcal{A}_1$. Otherwise, $\mathcal{B}$ uniformly selects an element $H' \in \mathbb{G}$, returns it and records the pair (X', H') in L_3.
- **Phase 1.** For a series of inquiries raised by $\mathcal{A}_1$, $\mathcal{B}$ answers as follows.

- PPKQuery oracle $\mathcal{O}_{ppk}$: $\mathcal{A}_1$ publishes an identity $PUID$ for query, $\mathcal{B}$ maintains a list of the form $L_{ppk}(PUID, D_{PUID})$ as the answer to $\mathcal{A}_1$. If $PUID$ already exists in the L_{ppk}, $\mathcal{B}$ first performs the H_1 Query in the above steps to obtain $H_1(PUID)$. Otherwise, $\mathcal{B}$ finds the tuple in the L_1:
  - Case1: If $|\omega \cap \omega_\pi| \geq d$, the challenger $\mathcal{B}$ aborts and outputs "fault".
  - Case2: If $|\omega \cap \omega_\pi| < d$, $\mathcal{B}$ randomly selects a polynomial $p(x)$ of degree $d-1$, $p(0) = a$, let $p(\omega_i) = a_i$, where $i \in \{1, \ldots, n\}$. $\mathcal{B}$ returns $z_i aP$ to $\mathcal{A}_1$, and saves tuple $(PUID, (D_{PUID}))$ in the L_{ppk}.
- PKQuery oracle $\mathcal{O}_{pk}$: $\mathcal{A}_1$ publishes an identity $PUID$ for query, $\mathcal{B}$ maintains a list of the form $L_{pub}(PUID, PK_{PUID}, (x_{i,PUID})_{i=1}^n)$ as the answer to $\mathcal{A}_1$. If $PUID$ already exists in the L_{pub}, $\mathcal{B}$ restores the corresponding record and returns PK_{PUID} to $\mathcal{A}_1$. Otherwise, $\mathcal{B}$ randomly selects $x_j \in Z_q^*$, a polynomial $y(x)$ of degree $d-1$, $y(0) = x_j$, let $y(\omega_i) = x_{i,PUID}$, where $i \in \{1, \ldots, n\}$. In this case, we suppose that $x_{PUID} = (x_{i,PUID})_{i=1}^n$ while $\mathcal{B}$ calculates $PK_{PUID} = x_{PUID}P$, and returns it to $\mathcal{A}_1$. Finally, $\mathcal{B}$ maintains $(PUID, PK_{PUID}, (x_{i,PUID})_{i=1}^n)$ in L_{pub}.
- PK replacement oracle $\mathcal{O}_{pkrp}$: When $\mathcal{A}_1$ queries the tuple $(PUID, \widetilde{PK_{PUID}})$, if $PUID$ has not been queried for the public key, $\mathcal{B}$ generates a public key query on $PUID$ to obtain PK_{PUID} and records $(PUID, \widetilde{PK_{PUID}}, \perp)$ in L_{pub}. Otherwise, $\mathcal{B}$ maintains $(PUID, \widetilde{PK_{PUID}}, \perp)$ in L_{pub}.
- SKQuery oracle $\mathcal{O}_{sk}$: $\mathcal{A}_1$ publishes an identity $PUID$ for query, $\mathcal{B}$ maintains a list of the form $L_{sk}(PUID, SK_{PUID})$ as the answer to $\mathcal{A}_1$. If $PUID$ has already queried, $\mathcal{B}$ restores the corresponding record and returns SK_{PUID} to $\mathcal{A}_1$, otherwise, $\mathcal{B}$ considers the following two cases:
  - Case 1: If $|\omega \cap \omega_\pi| \geq d$, $\mathcal{B}$ aborts and outputs "fault".
  - Case 2: If $|\omega \cap \omega_\pi| < d$, $\mathcal{B}$ returns the SK_{PUID} to $\mathcal{A}_1$ and saves tuple $(PUID, D_{PUID}, x_{PUID})$ in the L_{sk}.
- ReKeyGen oracle $\mathcal{O}_{rk}$: $\mathcal{B}$ first searches whether tuple $(PUID, PUID', RK_{S,\omega,\mathcal{R}})$ exists in the L_rk. If so, $\mathcal{B}$ returns $RK_{S,\omega,\mathcal{R}}$ to $\mathcal{A}_1$. Otherwise, we suppose that $\mathcal{A}_1$ has conducted the above series of queries when querying the ROM, so when $|\omega \cap \omega_\pi| \geq d$, $\mathcal{B}$ will follow the steps below:
  - Case 1: When $\alpha_1 = 1$, $\mathcal{B}$ follows the above steps to obtain $PUID$'s public–private key pair (SK_{PUID}, PK_{PUID}), and the public key PK'_{PUID} of $PUID'$. Then, $\mathcal{B}$ calculates $\varphi = e(D_{PUID}, H_1(PUID'))$, and the re-encryption key $RK_{S,\omega,\mathcal{R}} = -D_{PUID_j} - x_{PUID_j}E_j + H_3(\varphi \| x_{PUID_j} PK_{PUID'_j} \| \omega \| \omega')$.
  - Case 2: When $\alpha_1 = 0$ and $\alpha_2 = 1$, $\mathcal{B}$ response fails.
  - Case 3: When $\alpha_1 = 0$ and $\alpha_2 = 0$, $\mathcal{B}$ randomly selects $RK_{S,\omega,\mathcal{R}} \in \mathbb{G}$ and returns to $\mathcal{A}_1$.
- Re-encryption oracle $\mathcal{O}_{reen}$: Suppose that the public key of $PUID$ has not been replaced, the original ciphertext $CT = (U_1, V_1)$ at this time.
  - Case 1: If $|\omega \cap \omega_\pi| < d$, $\mathcal{B}$ aborts and outputs "fault".
  - Case 2: If $|\omega \cap \omega_\pi| \geq d$, $\mathcal{B}$ considers the following two cases:
    (1) If $\alpha_u = 1$, $\mathcal{B}$ aborts and outputs "fault".
    (2) If $\alpha_u = 0$, $\mathcal{B}$ re-encrypts the CT into $CT' = (U_1, V_1 e(U_1, RK_{S,\omega,\mathcal{R}}))$ and sends it to $\mathcal{A}_1$.

- **Challenge.** $\mathcal{A}_1$ outputs $PUID_\pi$ and two messages of equal length (m_0, m_1). If the flag variable $\alpha_u \neq 0$ of the challenge identity $PUID_\pi$, $\mathcal{B}$ fails in this game. Otherwise, $\mathcal{B}$ randomly selects a message m_b, where $b \in \{0, 1\}$, calculates the ciphertext $CT_b = (U_b, V_b) = (bP, m_b \prod_{\omega_i \in S} e(PK_{PUID_\pi}, t_i bP) T^{\Delta_{\omega_i, S}(0)})$ and sends CT_b to $\mathcal{A}_1$.
- **Phase 2.** Adversary $\mathcal{A}_1$ initiates a series of queries similar to Phase 1, and $\mathcal{B}$ responds accordingly. Please note that the queries issued by $\mathcal{A}_1$ in this phase must comply with the constraints in the security model.
- **Guess.** Once the adversary $\mathcal{A}_1$ provides a guess $b' \in \{0, 1\}$ for the challenge bit, $\mathcal{B}$ outputs 1 if $b' = b$ and 0 otherwise. □

Theorem 2. *If adversary $\mathcal{A}_2$ breaks FCL-PRE with a non-negligible advantage ε, we can construct an algorithm $\mathcal{B}$ that solves the DBDH assumption in polynomial time with an advantage ε'.*

Proof. Similar to the Theorem 1, therefore, only their main differences are presented below.

- **Initialization.** $\mathcal{B}$ returns the *params* and $msk = s$ to $\mathcal{A}_2$. It should be noted that $\mathcal{A}_2$ represents the KGC, which has access to the partial private key and is computed by challenger $\mathcal{B}$. Therefore, in this case, there is no need to simulate the PartialPrivateKey algorithm as well as the hash function H_1. Next, $\mathcal{B}$ randomly chooses an integer $r \in [1, q_{H_2}]$ and to the queries raised by $\mathcal{A}_2$, $\mathcal{B}$ answers as follows:
  - H_2 Query: When $\mathcal{A}_2$ queries the existing $PUID$ in L_2, $\mathcal{B}$ will respond with Y_i, otherwise it considers the following two situations:
    - Case 1: If $j = r$, $\mathcal{B}$ computes $H_2(PUID_j \| PK_{PUID_j} \| P_{pub}) = cP$ and returns it to $\mathcal{A}_2$.
    - Case 2: If $j \neq r$, $\mathcal{B}$ randomly selects $t_i \in Z_q^*$, and calculates $Y_i = t_i P$, then $\mathcal{B}$ returns it to $\mathcal{A}_2$. Finally, $\mathcal{B}$ adds the tuple $(PUID, t_i, Y_i)$ to L_2.
- **Phase 1.** For a series of inquiries raised by $\mathcal{A}_2$, $\mathcal{B}$ answers as follows.
  - PKQuery oracle $\mathcal{O}_{pk}$: $\mathcal{A}_2$ publishes an identity $PUID$ for query, $\mathcal{B}$ first selects $\pi \in [1, q_{pub}]$ randomly, and defines $PUID_\pi$ as the challenge identity.
    - Case 1: If $PUID$ has been queried, $\mathcal{B}$ restores the corresponding record and returns $PK_{PUID} = x_{PUID}P$ to $\mathcal{A}_2$.
    - Case 2: If $PUID$ has not been queried, then $\mathcal{B}$ considers the following scenario:
      (1) If $|\omega \cap \omega_\pi| < d$ and $j \neq \pi$, $\mathcal{B}$ selects a random number $x^*_{i,PUID_j} \in Z_q^*$, a polynomial $y(x)$ of degree $d-1$, $y(0) = x^*_{i,PUID_j}$, let $y(\omega_i) = x^*_{i,PUID_j}$, where $i \in \{1, \ldots, n\}$. Next, $\mathcal{B}$ calculates $PK_{PUID} = x_{PUID}P$, and returns it to $\mathcal{A}_2$. Finally, $\mathcal{B}$ saves the tuple $(PUID, (x_{i,PUID_j})_{i=1}^n, PK_{PUID})$ to L_{pub}.
      (2) If $|\omega \cap \omega_\pi| \geq d$ and $j = \pi$, $\mathcal{B}$ calculates $PK_{PUID} = aP$, and returns it to the adversary $\mathcal{A}_2$. Finally, $\mathcal{B}$ maintains the tuple $(PUID_\pi, (x_{i,PUID_j})_{i=1}^n, PK_{PUID})$ to the L_{pub}.
  - SKQuery oracle $\mathcal{O}_{sk}$: $\mathcal{B}$ considers the following two cases:
    - Case 1: If $PUID$ has been queried, $\mathcal{B}$ restores the corresponding record and returns SK_{PUID} to $\mathcal{A}_2$.
    - Case 2: If $PUID$ has not been queried, $\mathcal{B}$ considers the following scenario:

(1) If $|\omega \cap \omega_\pi| < d$ and $j \neq r$, $\mathcal{B}$ makes sure that $\mathcal{A}_2$ has performed PKQuery and all hash queries. Then, $\mathcal{B}$ calculates D_{PUID} and returns the $SK_{PUID} = (D_{PUID}, x_{PUID})$ to $\mathcal{A}_2$, while saving the tuple $(PUID, D_{PUID}, x_{PUID})$ in the L_{sk}.
(2) If $|\omega \cap \omega_\pi| \geq d$ and $j = r$, $\mathcal{B}$ aborts and outputs "fault".

– ReKeyGen oracle $\mathcal{O}_{rk}$: For the re-encryption key queries of $PUID$ and $PUID'$, when $|\omega \cap \omega_\pi| \geq d$, $\mathcal{B}$ makes the following answer:

(1) If $j \neq r$, the challenger $\mathcal{B}$ outputs the re-encryption key $RK_{S,\omega,\mathcal{R}} = -D_{PUID_j} - x_{PUID_j} E_j + H_3(\varphi \| x_{PUID_j} PK_{PUID'_j} \| \omega \| \omega')$.
(2) If $j = r$ and the private key of $PUID'$ has been queried, $\mathcal{B}$ responds with failure.
(3) If $j = r$ and the private key of $PUID'$ has not been queried, $\mathcal{B}$ randomly selects $RK_{S,\omega,\mathcal{R}} \in \mathbb{G}$ as the answer and returns it to $\mathcal{A}_2$.

- **Challenge.** $\mathcal{A}_2$ outputs $PUID_\pi$ and two messages of equal length (m_0, m_1). If the challenge identity $PUID_\pi \neq PUID_r$, $\mathcal{B}$ fails in this game. Otherwise, $\mathcal{B}$ randomly selects a message m_b, where $b \in \{0, 1\}$, calculates the ciphertext $CT_b = (U_b, V_b) = (bP, m_b \prod_{\omega_i \in S} e(bP, sH_1(PUID_\pi))T^{\Delta_{\omega_i,S}(0)})$ and sends CT_b to $\mathcal{A}_2$. □

6.2. Security properties of FCL-PRE

- *Confidentiality.* According to the above security proof, the proposed FCL-PRE scheme satisfies IND-CPA secure in the random oracle model and holds under the DBDH assumption. In addition, before re-encryption, the proxy CPS needs to authenticate registered users, and re-encryption is only allowed when the original ciphertext meets a certain condition, which further enhances the confidentiality of the scheme.
- *Anonymity.* FCL-PRE converts each user's real biometric identity UID_j into a pseudo-identity $PUID_j = Enc_\phi(UID_j \| n_{PUID_j}) \| T_j$ through a symmetric encryption algorithm for hiding. Therefore, if an adversary wishes to obtain UID_j, he/she must first acquire the symmetric key ϕ. However, in our scheme, only a trusted TA can extract ϕ, thereby ensuring the anonymity of the user's real identity.
- *Error tolerance.* We employ secret sharing technology to divide the system master key s and the secret value x_{PUID_j} into n independent components. Based on these components, the sender S_j generates the final complete private key and the corresponding ciphertext. In the verification phase, the ciphertext can be re-encrypted if the attribute set contains at least d valid attributes. Here, d is defined as an error tolerance parameter, so as to achieve the system's error tolerance and enhance its robustness.
- *Collusion Resistance.* Given the commercial nature of cloud service providers, a potential risk arises that they may collude with the receiver $\mathcal{R}_j$ to acquire S_j's private key $SK_{PUID_j} = (D_{PUID_j}, x_{PUID_j})$. However, under the threshold secret sharing, collusion between $\mathcal{R}_j$ and CPS is infeasible. First, S_j's full private key consists of a partial private key D_{PUID_j} and a secret value x_{PUID_j}, both of which are divided into n components. This means that at least t attribute shards must be obtained to recover one of the keys. Second, even if the colluder obtains x_{PUID_j}, they cannot deduce the sender's partial private key D_{PUID_j}, because $D_{PUID_j} = sH_1(PUID_j)$, where s is the master key. Since the master key s is unknown to the colluder, they cannot calculate D_{PUID_j}.

7. Performance evaluation

This section provides a systematic performance evaluation of FCL-PRE and other related schemes from both theoretical and experimental perspectives. First, we built an experimental system on Ubuntu 20.10, using Python 3.10 and Sagemath 9.8, setting the security parameter to $\lambda = 256$. The chosen elliptic curve E/F_p is defined by the simplified Weierstrass equation $y^2 = x^3 + ax + b$.

7.1. Theoretical analysis

Table 3 compares the number of modular exponentiations, scalar multiplications, and bilinear pairings for FCL-PRE, YDKR21 [43], FLWL24 [24], and ZZYL20 [44], to assess the computational overhead at different stages. All three references adopt CL-PRE in data-sharing scenarios. In the following, we focus on the major computational overhead on the sender side S_j.

Encryption: The efficiency ranking is YDKR21 [43] < FLWL24 [24] < Ours < ZZYL20 [44]. Since biometric characteristic *bio* inevitably contains noise during collection, FCL-PRE binds each registered user's pseudo-identity to an attribute set $\{\omega\}_{i=1}^n$. Consequently, during encryption, S_j must bind attribute fragments to the message, ensuring both data confidentiality and system error tolerance.

ReKey Generation: The efficiency ranking is YDKR21 [43] < ZZYL20 [44] < Ours < FLWL24 [24]. In FCL-PRE, users are allowed to omit or update some attributes during key generation, eliminating the extra computational overhead associated with regenerating public–private key pairs. Moreover, even if the proxy CPS colludes with the receiver, it cannot deduce the user's real identity from the re-encryption key.

Decrypt1: The efficiency ranking is ZZYL20 [44] < YDKR21 [43] < FLWL24 [24] = Ours. Compared to ZZYL20 [44] and YDKR21 [43], FCL-PRE improves the decryption efficiency on the sender side S_j by 40.57% and 44.6%, respectively, significantly reducing computational burden.

In summary, by integrating certificateless encryption with secret sharing technology, FCL-PRE enhances user privacy and system error tolerance while effectively addressing the stringent privacy requirements in cloud-based data-sharing scenarios.

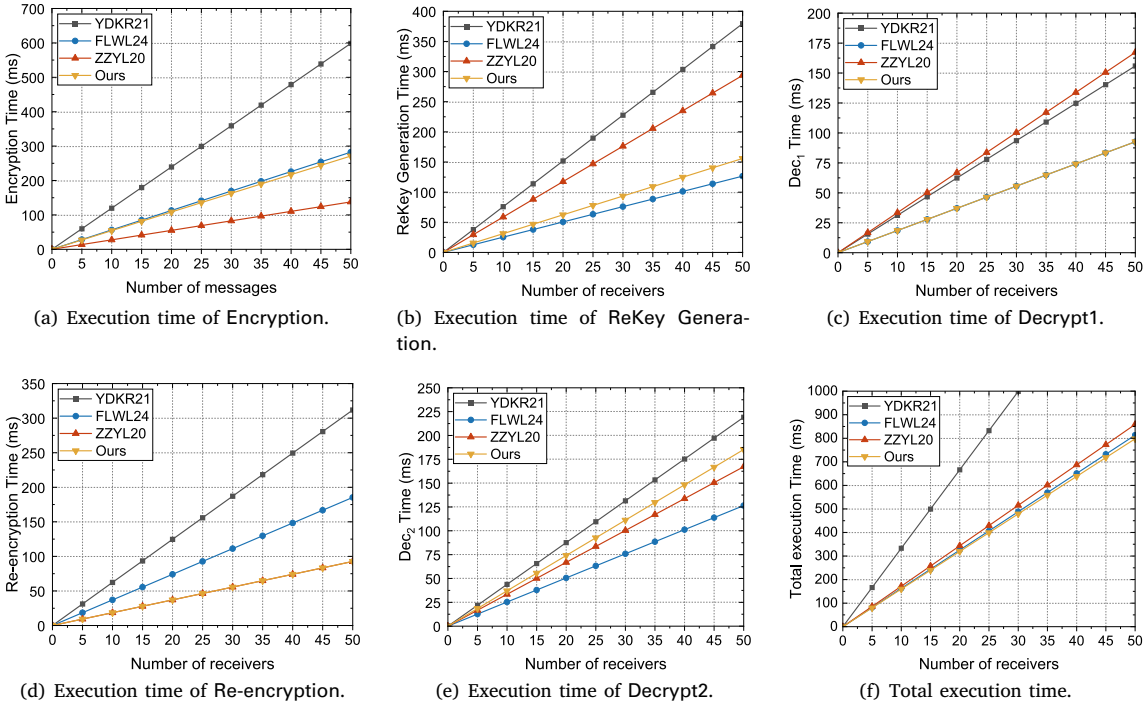
7.2. Experimental analysis

Computational overhead. To ensure the objectivity and accuracy of our results, we excluded the Setup algorithm from the experiment, as it is executed only once and has a negligible impact on the user encryption experience. For the remaining algorithms, each was executed 100 times, and the average execution time was recorded. Fig. 4 reports the execution time of all main stages in our scheme as a function of the number of receivers/messages. Specifically, Fig. 4(a)–(c) show the sender-side costs, including Encryption time, ReKey Generation time, and Decrypt1 time, respectively. Fig. 4(d) presents the Re-encryption time at the cloud proxy server, while Fig. 4(e) depicts the Decrypt2 time at the authorized receiver. Fig. 4(f) summarizes the total computational overhead across all parties. As the number of receivers/messages increases, all stages exhibit an approximately linear growth. Our FCL-PRE scheme consistently incurs lower decryption time, re-encryption time, and overall computational cost than the compared schemes, as illustrated in Fig. 4(c), (d), and (f). These results demonstrate that FCL-PRE achieves better efficiency and scalability, particularly in multi-receiver settings.

Communication overhead. Table 3 compares the communication overhead of YDKR21 [43], FLWL24 [24], ZZYL20 [44], and our proposed scheme. The storage and transmission overheads of the data sender and cloud proxy server, including the original ciphertext, re-encryption key, and re-encrypted ciphertext, are discussed in detail.

Table 3
Comparison of cryptographic operations of related schemes.

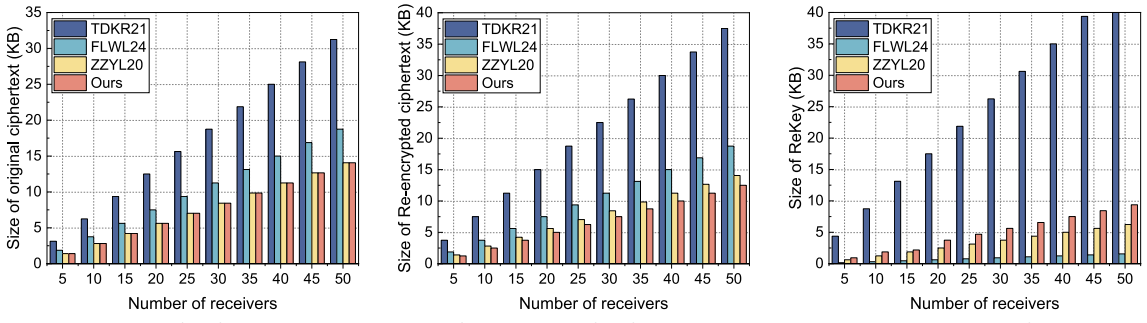
Scheme	Computational cost					Communication cost		
	Encryption	ReKeyGen	Re-encryption	Decrypt1	Decrypt2	CT_1	CT_2	ReKey
YDKR21 [43]	$T_p + 8T_e$	$6T_e$	$2T_p + 2T_e$	$T_p + T_e$	$T_p + 2T_e$	$3\|\mathbb{G}\| + 2\|\mathbb{G}_T\|$	$4\|\mathbb{G}\| + 2\|\mathbb{G}_T\|$	$6\|\mathbb{G}\| + 4\|Z_q^*\|$
FLWL24 [24]	$T_p + 3T_e$	$2T_e$	$2T_p$	T_p	$2T_e$	$2\|\mathbb{G}\| + \|\mathbb{G}_T\|$	$3\|\mathbb{G}_T\|$	$\|\mathbb{G}\|$
ZZYL20 [44]	$2T_e + T_{sm}$	$T_p + 3T_e + T_{sm}$	T_p	$T_p + T_e + T_{sm}$	$T_p + T_e + T_{sm}$	$2\|\mathbb{G}\| + \|Z_q^*\|$	$2\|\mathbb{G}\| + \|Z_q^*\|$	$\|Z_q^*\|$
Ours	$2T_p + T_e + 2T_{sm}$	$T_p + T_e$	T_p	T_p	$2T_p$	$\|\mathbb{G}\| + \|\mathbb{G}_T\| + \|Z_q^*\|$	$\|\mathbb{G}\| + \|\mathbb{G}_T\|$	$\|\mathbb{G}\| + 2\|Z_q^*\|$



(a) Execution time of Encryption. (b) Execution time of ReKey Generation. (c) Execution time of Decrypt1.

(d) Execution time of Re-encryption. (e) Execution time of Decrypt2. (f) Total execution time.

Fig. 4. The execution time of each phase.



(a) Original ciphertext. (b) Re-encrypted ciphertext. (c) Re-encryption key.

Fig. 5. Communication overhead comparison.

Sender side: Regarding the transmission of the original ciphertext, our proposed scheme and ZZYL20 [44] achieve the lowest communication cost, as shown in Fig. 5(a). Although our scheme incurs slightly higher communication overhead for the transmission of the re-encryption key compared to ZZYL20 [44], it is worth noting that ZZYL20 pre-generates and stores the re-encryption key in the cloud, which may lead to a potential risk of key misuse. As we can see in Fig. 5(c), FCL-PRE requires only KB level for storage, making it well-suited for resource-constrained mobile devices without imposing a significant burden on the sender side.

Cloud proxy server (CPS) side: For the storage of re-encrypted ciphertext, our scheme also demonstrates the lowest communication cost, as

shown in Fig. 5(b). Even when the number of designated recipients is relatively large, i.e., 50 receivers, FCL-PRE requires only 12.5 KB of communication overhead at the CPS side. It indicates that FCL-PRE not only effectively minimizes the cloud's communication burden but also ensures a flexible and reliable sharing mechanism without compromising data security.

8. Conclusion

In this paper, we propose FCL-PRE, a fuzzy certificateless proxy re-encryption scheme that facilitates flexible key management while ensuring efficient and secure data sharing. By integrating anonymous biometric recognition, our approach conceals users' real identities, achieving effective conditional privacy and bolstering system error tolerance. Notably, we prevent malicious re-encryption requests by verifying the signature, while secret sharing technology enhances collusion resistance. Moreover, a formal security analysis under the random oracle model demonstrates that FCL-PRE resists chosen-plaintext attacks. Compared to existing schemes, FCL-PRE significantly reduces computational and communication overhead, achieving the lowest total computational cost and ciphertext storage overhead. In future work, we aim to optimize dynamic user revocation and enhance adaptability to real-world cloud environments with more complex access policies.

CRediT authorship contribution statement

Jiasheng Chen: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Zhenfu Cao:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Liangliang Wang:** Writing – review & editing, Validation, Methodology, Formal analysis, Data curation. **Jiachen Shen:** Validation, Supervision, Formal analysis. **Xiaolei Dong:** Validation, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62132005, 62172162), in part by Shanghai Trusted Industry Internet Software Collaborative Innovation Center, in part by Fundamental Research Funds for the Central Universities, in part by Police Integration Computing Key Laboratory of Sichuan Province (Grant No. JWRH202401001).

Data availability

Data will be made available on request.

References

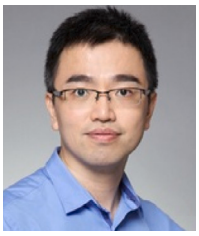
[1] Shuzhou Sun, Hui Ma, Zishuai Song, Rui Zhang, WebCloud: Web-based cloud storage for secure data sharing across platforms, IEEE Trans. Dependable Secur. Comput. 19 (3) (2020) 1871–1884.
[2] Maithilee Joshi, Karuna P. Joshi, Tim Finin, Delegated authorization framework for ehr services using attribute-based encryption, IEEE Trans. Serv. Comput. 14 (6) (2019) 1612–1623.
[3] Yinbin Miao, Robert H. Deng, Ximeng Liu, Kim-Kwang Raymond Choo, Hongjun Wu, Hongwei Li, Multi-authority attribute-based keyword search over encrypted cloud data, IEEE Trans. Dependable Secur. Comput. 18 (4) (2019) 1667–1680.
[4] Matt Blaze, Gerrit Bleumer, Martin Strauss, Divertible protocols and atomic proxy cryptography, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 1998, pp. 127–144.
[5] Matthew Green, Giuseppe Ateniese, Identity-based proxy re-encryption, in: Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007, Springer, 2007, pp. 288–306.
[6] Chunpeng Ge, Willy Susilo, Jiandong Wang, Liming Fang, Identity-based conditional proxy re-encryption with fine-grained policy, Comput. Stand. Interfaces 52 (2017) 1–9.
[7] Hongmei Pei, Peng Yang, Weihao Li, Miao Du, Zhongjian Hu, Proxy re-encryption for secure data sharing with blockchain in internet of medical things, Comput. Netw. 245 (2024) 110373.
[8] Guijiang Liu, Haibo Xie, Wenming Wang, Haiping Huang, A secure and efficient electronic medical record data sharing scheme based on blockchain and proxy re-encryption, J. Cloud Comput. 13 (1) (2024) 44.
[9] Anca-Andreea Ivan, Yevgeniy Dodis, Proxy cryptography revisited, in: NDSS, 2003.
[10] Yang Lu, Efficient certificate-based proxy re-encryption scheme for data sharing in public clouds, KSII Trans. Internet Inf. Syst. (TIIS) 9 (7) (2015) 2703–2718.
[11] Zhiguang Qin, Hu Xiong, Shikun Wu, Jennifer Batamuliza, A survey of proxy re-encryption for secure data sharing in cloud computing, IEEE Trans. Serv. Comput. (2016) 1–18.
[12] Giuseppe Ateniese, Kevin Fu, Matthew Green, Susan Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM Trans. Inf. Syst. Secur. (TISSEC) 9 (1) (2006) 1–30.
[13] Craig Gentry, Certificate-based encryption and the certificate revocation problem, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2003, pp. 272–293.
[14] Chul Sur, Youngho Park, Sang Uk Shin, Kyung Hyune Rhee, Changho Seo, Certificate-based proxy re-encryption for public cloud storage, in: 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE, 2013, pp. 159–166.
[15] Chunpeng Ge, Zhe Liu, Jinyue Xia, Liming Fang, Revocable identity-based broadcast proxy re-encryption for data sharing in clouds, IEEE Trans. Dependable Secur. Comput. 18 (3) (2019) 1214–1226.
[16] Jing Zhang, Shuangshuang Su, Hong Zhong, Jie Cui, Debiao He, Identity-based broadcast proxy re-encryption for flexible data sharing in VANETs, IEEE Trans. Inf. Forensics Secur. 18 (2023) 4830–4842.
[17] Jiguo Li, Xuexia Zhao, Yichen Zhang, Certificate-based conditional proxy re-encryption, in: International Conference on Network and System Security, Springer, 2015, pp. 299–310.
[18] Jun Shao, Peng Liu, Yuan Zhou, Achieving key privacy without losing CCA security in proxy re-encryption, J. Syst. Softw. 85 (3) (2012) 655–665.
[19] Jian Weng, Robert H. Deng, Xuhua Ding, Cheng-Kang Chu, Junzuo Lai, Conditional proxy re-encryption secure against chosen-ciphertext attack, in: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, 2009, pp. 322–332.
[20] Cui Li, Rongmao Chen, Yi Wang, Qianqian Xing, Baosheng Wang, REEDS: An efficient revocable end-to-end encrypted message distribution system for IoT, IEEE Trans. Dependable Secur. Comput. 21 (5) (2024) 4526–4542.
[21] Shimao Yao, Ralph Voltaire J. Dayot, In-Ho Ra, Liya Xu, Zhuolin Mei, Jiaoli Shi, An identity-based proxy re-encryption scheme with single-hop conditional delegation and multi-hop ciphertext evolution for secure cloud data sharing, IEEE Trans. Inf. Forensics Secur. 18 (2023) 3833–3848.
[22] Giuseppe Ateniese, Karyn Benson, Susan Hohenberger, Key-private proxy re-encryption, in: Cryptographers' Track at the RSA Conference, Springer, 2009, pp. 279–294.
[23] Chengdong Ren, Xiaolei Dong, Jiachen Shen, Zhenfu Cao, Yuanjian Zhou, Clappre: Certificateless autonomous path proxy re-encryption for data sharing in the cloud, Appl. Sci. 12 (9) (2022) 4353.
[24] Jingyu Feng, Yue Li, Teng Wang, Shuanggen Liu, A certificateless threshold proxy re-encrypted data sharing scheme with cloud-chain collaboration in industrial internet environments, IEEE Internet Things J. 11 (20) (2024) 33247–33268.
[25] Liqing Chen, Meng Zhang, Jiguo Li, Conditional identity-based broadcast proxy re-encryption with anonymity and revocation, IEEE Trans. Reliab. 74 (3) (2025) 3573–3584.
[26] Liming Fang, Jiandong Wang, Chunpeng Ge, Yongjun Ren, Fuzzy conditional proxy re-encryption, Sci. China Inf. Sci. 56 (5) (2013) 1–13.
[27] BaoHong Li, JieFei Xu, YanZhi Liu, Lattice-based fuzzy conditional proxy re-encryption, J. Internet Technol. 20 (5) (2019) 1379–1385.
[28] Binhan Li, Lunzhi Deng, Yiming Mou, Na Wang, Yanli Chen, Siwei Li, A pairing-free data sharing scheme based on certificateless conditional broadcast proxy re-encryption suitable for cloud-assisted IoT, IEEE Internet Things J. 12 (20) (2025) 42754–42768.
[29] Yousheng Zhou, Yurong Li, Yuanni Liu, A certificateless and dynamic conditional proxy re-encryption-based data sharing scheme for IoT cloud, J. Internet Technol. 26 (2) (2025) 165–172.
[30] Shi Lin, Li Cui, Niu Ke, End-to-end encrypted message distribution system for the Internet of Things based on conditional proxy re-encryption, Sensors 24 (2) (2024) 1–16.
[31] Yongjing Zhang, Zhouyang Zhang, Shan Ji, Shenqing Wang, Shitao Huang, Conditional proxy re-encryption-based key sharing mechanism for clustered federated learning, Electronics 13 (5) (2024) 848.

[32] Chul Sur, Chae Duk Jung, Youngho Park, Kyung Hyune Rhee, Chosen-ciphertext secure certificateless proxy re-encryption, in: IFIP International Conference on Communications and Multimedia Security, Springer, 2010, pp. 214–232.
[33] Sattam S. Al-Riyami, Kenneth G. Paterson, Certificateless public key cryptography, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2003, pp. 452–473.
[34] Tarunpreet Bhatia, Anil K. Verma, Gaurav Sharma, Secure sharing of mobile personal healthcare records using certificateless proxy re-encryption in cloud, Trans. Emerg. Telecommun. Technol. 29 (6) (2018) e3309.
[35] Nabeil Eltayieb, Liang Sun, Ke Wang, Fagen Li, A certificateless proxy re-encryption scheme for cloud-based blockchain, in: Frontiers in Cyber Security: Second International Conference, FCS 2019, Xi'an, China, November 15–17, 2019, Proceedings 2, Springer, 2019, pp. 293–307.
[36] Emmanuel Ahene, Junfeng Dai, Hao Feng, Fagen Li, A certificateless signcryption with proxy re-encryption for practical access control in cloud-based reliable smart grid, Telecommun. Syst. 70 (2019) 491–510.
[37] Amit Sahai, Brent Waters, Fuzzy identity-based encryption, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2005, pp. 457–473.
[38] Hu Xiong, YaNan Chen, GuoBin Zhu, ZhiGuang Qin, Analysis and improvement of a provable secure fuzzy identity-based signature scheme, Sci. China Inf. Sci. 57 (2014) 1–5.
[39] Liangliang Wang, Jiangwei Xu, Baodong Qin, Mi Wen, Kefei Chen, An efficient fuzzy certificateless signature-based authentication scheme using anonymous biometric identities for VANETs, IEEE Trans. Dependable Secur. Comput. 22 (1) (2024) 292–307.
[40] Dan Boneh, Matt Franklin, Identity-based encryption from the Weil pairing, in: Annual International Cryptology Conference, Springer, 2001, pp. 213–229.
[41] Adi Shamir, How to share a secret, Commun. ACM 22 (11) (1979) 612–613.
[42] A. Riyami, Sattam S., K.G. Paterson, Certificateless public key cryptography, in: Chi-Sung Laih (Ed.), Advances in Cryptology - ASIACRYPT 2003, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 452–473.
[43] Shimao Yao, Ralph Voltaire J. Dayot, Hyung-Jin Kim, In-Ho Ra, A novel revocable and identity-based conditional proxy re-encryption scheme with ciphertext evolution for secure cloud data sharing, IEEE Access 9 (2021) 42801–42816.
[44] Xiaoyu Zheng, Yuyang Zhou, Yalan Ye, Fagen Li, A cloud data deduplication scheme based on certificateless proxy re-encryption, J. Syst. Archit. 102 (2020) 101666.

Jiasheng Chen is currently pursuing the Ph.D. degree with the Department of Cryptography and Cyber Security School of Software Engineering, East China Normal University, Shanghai, China. Her research interests include applied cryptography and information security.

Zhenfu Cao is currently a Distinguished Professor with East China Normal University, China. Since 1981, he has been published over 400 academic papers in journals or conferences. His research interests include cryptography, number theory, and information security. He has received a number of awards, including the Ying-Tung Fok Young Teacher Award, in 1989, the National Outstanding Youth Fund of China, in 2002, and the Special Allowance by the State Council, in 2005. He was a co-recipient of the 2007 IEEE International Conference on Communications Computer Award, in 2007.

Liangliang Wang received the Ph.D. degree from Shanghai Jiao Tong University, in 2016. He has published academic papers in prestigious venues including IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Vehicular Technology, IEEE Internet of Things Journal, Knowledge-Based Systems and SCIENCE CHINA Information Sciences. He is currently an Associate Professor with the College of Computer Science and Technology, Shanghai University of Electric Power. His research interests include applied cryptography, information security and privacy preserving.



Jiachen Shen received the bachelor's degree from Shanghai Jiao Tong University, Shanghai, China, in 2001, and the master's and Ph.D. degrees from the University of Louisiana at Lafayette, Lafayette, LA, USA, in 2003 and 2008, respectively. He joined East China Normal University, Shanghai, China, in 2015. His research interests include applied cryptography, cloud security, searchable encryption, and blockchains.

Xiaolei Dong is currently a Distinguished Professor with East China Normal University. She hosts a lot of research projects supported by the National Basic Research Program of China (973 Program), the National Natural Science Foundation of China, and the Special Funds on Information Security of the National Development and Reform Commission. Her research interests include cryptography, number theory, and trusted computing.