Subversion-Resistant Autonomous Path Proxy Re-Encryption with Secure Deduplication for IoMT

Jiasheng Chen, Zhenfu Cao, *Senior Member, IEEE,* Lulu Wang, Jiachen Shen, *Member, IEEE,* Zehui Xiong, *Senior Member, IEEE,* and Xiaolei Dong, *Member, IEEE*

***Abstract*—The Internet of Medical Things (IoMT) consists of many resource-constrained medical devices that provide patients with medical services anytime and anywhere. In such an environment, the collection and sharing of medical records raise serious security concerns. Although various cryptographic schemes have been proposed, most fail to address two critical threats simultaneously: (i) sensitive data exposure caused by external cloud servers and/or open network environments; (ii) algorithm substitution attacks (ASAs) by internal adversaries. Furthermore, when data owners (e.g., delegators) are inconvenient to process their data, it is desirable to establish a more fine-grained way to delegate encryption rights. To tackle these issues, we propose a subversion-resistant autonomous path proxy re-encryption with an equality test function (SRAP-PRET). Specifically, our scheme allows the delegator to create a multi-hop delegation path in advance with the delegator's preferences, effectively preventing unauthorized access. By deploying a cryptographic reverse firewall zone, SRAP-PRET addresses the problem of information leakage caused by adversaries initiating ASAs. Additionally, SRAP-PRET also supports secure deduplication and efficient data decryption. Security analysis shows that SRAP-PRET provides resistance against ASAs and security against chosen plaintext attacks. Performance evaluations demonstrate that SRAP-PRET offers enhanced security properties without sacrificing efficiency.**

***Index Terms*—Proxy re-encryption, autonomous path delegation, subversion-resistant, deduplication, IoMT security.**

I. Introduction

THE rapid development of IoMT has profoundly changed the healthcare industry by enabling real-time data collection, analysis, and sharing among interconnected medical devices. In practice, IoMT usually integrates different types of devices, including wearable devices, medical sensors, and hospital servers. These devices jointly aggregate clinical data to construct electronic health records (EHRs) [1], which provide valuable decision support for healthcare institutions. However, given the highly sensitive nature of EHRs, any unauthorized access or data leakage raises serious privacy risks and may even endanger the operation of life-support devices [2]. According to the IBM 2023 Healthcare Data report [3], 82% of healthcare organizations experienced an IoT-focused cyber attack last year. These medical devices introduce unique security vulnerabilities that may compromise patient confidentiality and endanger safety if exploited. Meanwhile, differences in computational power and energy constraints across these devices complicate the implementation of robust security protocols. As a result, securely and efficiently sharing EHRs across organizations has become a critical challenge for enabling effective collaboration among medical staff while preserving patient privacy.

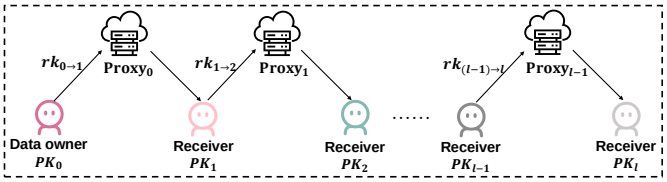


Fig. 1: Multi-hop property of PRE.

Proxy re-encryption (PRE) provides a promising approach to addressing this challenge. Its fundamental idea is that a semi-trusted proxy, once provided with a re-encryption key, can transform a ciphertext encrypted to one recipient can be converted to another authorized recipient's ciphertext without plaintext disclosure [4]. The property that allows a delegator to achieve cross-organizational data sharing with multiple authorized institutions by generating the corresponding re-encryption keys is called multi-hop [5], as shown in Fig.1. Through these abilities, PRE not only improves privacy but also facilitates cross-organizational data accessibility. Hence, it is recognized as an effective strategy for safeguarding sensitive medical information in IoMT environments [6]–[10].

A. Motivation and Challenge

Although various flexible PRE technologies bring significant advantages, existing schemes still face the following challenges when dealing with the complex requirements:

Trust-weakening risk of multi-hop PRE. IoMT data-sharing requirements are often dynamic, prompting the need for multi-hop PRE [11]–[13]. This technique allows proxies to re-encrypt data without decrypting the original ciphertext, facilitating real-time updates of access permissions. For example, the hospital $\mathscr{A}$ authorizes a specialist Bob to access patient Alice's EHRs. Suppose Bob is unable to make a diagnosis in time due to a business trip or being offline. In that case, he can further delegate the decryption authority to Carol, another specialist at the hospital. Under single-hop PRE [14]–[16], such a transfer of permission is difficult to achieve, whereas multi-hop PRE allows this process to be carried out with

Jiasheng Chen, Zhenfu Cao, Jiachen Shen, and Xiaolei Dong are with the Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China. Zhenfu Cao is the corresponding author (Email: zfcao@sei.ecnu.edu.cn).

Lulu Wang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China.

Zehui Xiong is with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, United Kingdom.

greater flexibility. Nevertheless, each additional "hop" risks weakening trust between the original delegator and the subsequent delegatees. Especially in cross-organizational scenarios, the extension of the delegation path will reduce the degree of control that data owners retain over their data. Consequently, strict safeguards are necessary to prevent both the misuse of permissions and the leakage of sensitive information through indirect authorization.

Internal threats of algorithm subversion under adversarial eavesdropping. Users rely on devices (such as computers or smart medical equipment) to correctly implement cryptographic algorithms for secure and convenient healthcare services. Yet even robust algorithms can be covertly subverted by internal adversaries, as highlighted by the "PRISM" and "CRYPTO AG" cases [17]. A backdoor in the algorithm implementation could allow attackers to steal keys or sensitive EHRs undetected. These attacks are particularly destructive because most encryption algorithms contain randomness, which attackers can exploit [18]. While deterministic encryption somewhat reduces randomness, it exposes the one-to-one correspondence between plaintext and ciphertext under the indistinguishability attack. Therefore, there is an urgent need for a PRE scheme that combines subversion resistance with confidentiality.

Efficiency and resource consumption in secure deduplication. In traditional encryption schemes [19], the same plaintext encrypted by different users produces distinct ciphertexts, leading to considerable storage overhead for cloud servers. Studies indicate that approximately 75% of digital data is redundant, and the redundancy rate in backup and archival systems may exceed 90% [20]. This issue becomes even more critical in the IoMT, where large data volumes must be stored and processed under the constraints of limited computational power and storage capacity on many devices. Users expect cloud-based data to remain secure and easily accessible, yet also seek optimal storage efficiency. Public key encryption with equality test (PKEET) [21] addresses this challenge by identifying ciphertexts that derive from the same plaintext, thus eliminating redundant data. However, current PKEET schemes usually depend on costly bilinear pairing operations. Such operations impose a heavy burden on resource-constrained IoMT nodes, particularly in multi-user settings where equality tests are frequently executed and further increase computational and energy consumption. These limitations highlight the need for more efficient deduplication methods that can preserve strong security guarantees while reducing resource overhead.

This work seeks to address the following critical question: *"Can we develop a flexible, fine-grained PRE scheme that resists ASAs while supporting efficient deduplication to accommodate diverse, multi-organizational IoMT needs?"*

B. Our Contributions

To tackle these challenges, we propose a proxy re-encryption scheme with autonomous path delegation, equality test, and subversion resistance characteristics, dubbed SRAP-PRET. First, SRAP-PRET supports ciphertext searches across diverse public key encryptions for efficient deduplication. Second, it enables delegators to customize the delegation path, thereby restricting proxy privileges for fine-grained decryption management. Finally, SRAP-PRET strengthens security and robustness against internal adversarial eavesdropping and data leakage from cloud servers or open networks. Specifically, the main contributions of this paper are as follows:

- **Autonomous path delegation and security data deduplication.** SRAP-PRET allows the delegator to establish a trusted authorization path, maintaining data ownership during multi-hop ciphertext re-encryption and preventing trust degradation. Moreover, it checks whether ciphertexts originate from the same plaintext, storing only the owner's ciphertext if duplicates arise. Authorized users then employ proxy re-encryption to access the data, thus forming a secure deduplication cycle.
- **Subversion-resistant cryptographic reverse firewall.** To avert subversion attacks under adversarial eavesdropping, SRAP-PRET deploys independent cryptographic reverse firewalls (CRFs) on both the PKG and the client, namely $\mathcal{W}_{\mathsf{PKG}}$ and $\mathcal{W}_{\mathsf{CL}}$, to form a jointly virtual trusted zone. By sanitizing keys and related messages through key malleability and element re-randomization, CRFs effectively prevent the establishment of hidden backdoor channels. Consequently, even if attackers use compromised medical devices, they cannot recover sensitive data from sanitized information.
- **Enhanced security without compromising performance.** We provide rigorous security proofs showing that SRAP-PRET achieves exfiltration resistance, indicating that attackers who obtain secret random values cannot deduce the underlying sensitive messages. Our theoretical analysis and experiments demonstrate that SRAP-PRET, even with CRFs in place, maintains only limited computational and communication overhead. This balance between security and efficiency makes it well-suited to multi-user settings.

II. Related Work

We review the literature on tackling the dual challenges of secure data sharing and efficient retrieval in multi-user environments, with a focus on proxy re-encryption, public-key encryption with equality test, and cryptographic reverse firewall (CRF). (1) PRE enables flexible delegation of decryption rights but lacks efficient support for ciphertext retrieval across different keys. (2) PKEET addresses this limitation by allowing servers to test ciphertext equality without exposing plaintexts, thereby facilitating cross-key search. Despite these advantages, both PRE and PKEET remain vulnerable to subversion attacks and mass surveillance. (3) CRFs provide a practical defense, ensuring confidentiality even in compromised environments.

Collectively, these three directions illustrate the progression of cryptographic techniques toward security sharing, retrieval, and resistance to subversion.

A. Proxy Re-Encryption

PRE is a public-key encryption scheme that enables a semi-trusted proxy to transform ciphertexts encrypted under one

user's public key into ciphertexts decryptable under another's key, without revealing the underlying plaintext. Since its introduction by Blaze et al. [22] at Eurocrypt'98, PRE has evolved into a popular technology for secure data sharing in multi-user environments.

(1) Classification and functional extensions of PRE. PRE schemes are typically classified along two dimensions: transformation depth and delegation direction.

- Single-hop vs. Multi-hop: Single-hop PRE allows only one transformation, whereas multi-hop PRE enables successive conversions among multiple users. When a delegatee is unavailable or overloaded, thus ensures flexible data accessibility and enhances scalability in distributed systems.
- Bidirectional vs. Unidirectional: Bidirectional schemes permit mutual re-encryption between two parties using the same re-encryption key, while unidirectional schemes restrict delegation to a one-way flow (from delegator to delegatee). The latter is generally preferred for practical access control, although it is technically more challenging to design.

Over time, a rich body of research [23]–[32] has extended PRE toward diverse functionalities, including revocation [27], accountability [25], searchability [32], and certificateless [29].

(2) Fine-grained and attribute-based PRE. Green and Ateniese [4] introduced identity-based PRE (IB-PRE), enabling flexible data sharing and simplifying key management. Nevertheless, once a proxy obtains a re-encryption key, it can convert all ciphertexts of the delegator for the delegatee. Therefore, a number of fine-grained PRE variants were proposed.

- Type-based PRE [33] and conditional PRE [34]–[37] restrict re-encryption to ciphertexts associated with certain categories or conditions.
- Attribute-based PRE (AB-PRE) [38]–[43] extends traditional PRE by combining attribute-based encryption (ABE) schemes, thereby supporting richer access structures and enabling more expressive delegation policies.

However, the heavy reliance on pairing and exponentiation operations in AB-PRE leads to substantial computational and storage costs, posing practical challenges for both private key generators and resource-constrained devices.

(3) Scalability and trust management in multi-hop PRE. To enhance scalability, multi-hop PRE [44]–[48] allows chained delegations, enabling ciphertext transformation through multiple proxies. However, in multi-user environments such as the IoMT, delegators typically have confidence only in their immediate delegatees, while trust decreases along longer delegation paths. To overcome this issue, Cao et al. [49] proposed an autonomous path PRE (AP-PRE), which enables the delegator to explicitly define an ordered path of delegatees and maintain tighter control over the delegation process. At the same time, as data outsourcing to cloud servers has become increasingly common, another fundamental challenge has emerged concerning the efficient retrieval of ciphertexts encrypted under heterogeneous keys.

In summary, PRE has developed from a simple delegation mechanism into a more complex framework that supports conditional access and multi-hop flexibility. Nevertheless, they still provide limited support for retrieval under multiple encryption keys.

B. Public-Key Encryption with Equality Test

Public key encryption with equality test (PKEET) allows a semi-trusted server to determine whether two ciphertexts, encrypted under different public keys, correspond to the same plaintext without revealing any information about the plaintext. This mechanism provides an effective means for secure data management and retrieval in multi-user cloud environments.

(1) Fundamental concept and mechanism of PKEET. The concept of PKEET was first introduced by Yang et al. [21], who demonstrated its potential for enabling equality testing directly over ciphertexts. The core idea of PKEET is straightforward. When two users, such as Alice and Bob, encrypt their respective data using different public keys and upload the ciphertexts to a cloud server, the server can execute an equality test algorithm to verify whether the ciphertexts correspond to the same underlying plaintext without performing decryption. Numerous public-key encryption with equality test schemes [58]–[65] have been developed for diverse application scenarios. For instance, Hassan et al. [62] presented a certificateless PKEET scheme that improves upon Elhabob's [63] construction by reducing the number of hash operations, thereby enhancing efficiency. Xu et al. [64] introduced a verifiable PKEET scheme designed for 5G networks, enabling users to perform reliable searches over massive datasets. Chen et al. [61] proposed a lightweight public key encryption with similarity test (PKEST) for electronic medical record (EMR) classification. PKEST can resist offline message recovery attacks that may be launched by an insider manager and eliminates the need for traditional pairing computations. In addition, this property also allows the cloud to conduct secure data deduplication, an operation that is especially important in large-scale IoT environments where lots of devices continuously generate redundant data. Through equality testing, the cloud server can identify and eliminate duplicate ciphertexts, storing only one copy to save cloud storage resources while significantly maintaining data privacy.

(2) Limitations of traditional PKEET. Traditional PKEET schemes still suffer from inherent limitations that hinder their adoption in IoMT.

- First, although equality testing can detect duplicate ciphertexts, the stored ciphertext remains bound to the original owner's public key. Consequently, if the cloud only retains Alice's ciphertext, even if two users have the same data, Bob cannot decrypt it using his own private key, which also complicates secure data sharing and recovery.
- Second, in the cloud storage, the conventional "search–download–decrypt–encrypt" workflow typically requires the data owner to re-encrypt data for each authorized user, causing high computational and communication overhead.

TABLE I: Summary of defense techniques against mass-surveillance subversion attacks.

Defense Techniques	Main Features	Strengths	Weaknesses			Complexity
			Scalability	Efficiency	Generality	
Multi-source Device Defense [50]	Multiple independent devices	High robustness	*Poor*	▼	–	▲
Unique Ciphertext Scheme [51], [52]	One-to-one ciphertext mapping	Deterministic mapping	●	▲	*Poor*	●
Unique Signature Scheme [53]	Deterministic signature	Easy verification	●	▲	*Poor*	▼
Decomposition–Amalgamation [54], [55]	Splitting and merging modules	Modular adaptability	●	▼	●	▲
Cryptographic Reverse Firewall [56]	Re-randomizes inputs/outputs	Leakage resilience	▲	●	▲	●
Self-guarding Protocol [57]	Trusted initialization ("anchor")	Autonomous checking	–	▼	*Poor*	●

Scalability: The ability of the defense scheme to maintain stable performance and security guarantees as the system size, number of participants, or data volume increases. **Efficiency:** The computational and communication efficiency of the algorithm during execution. **Generality:** The adaptability and reusability of the scheme across different cryptographic primitives or application scenarios without extensive modification.
"▲": High; "●": Medium; "▼": Low; "–": Not considered in the scheme.

(3) The emergence of proxy re-encryption with equality test. In 2010, a new concept called proxy re-encryption with keyword search (PRE-KS) was introduced by Shao et al. [66], which equips the search functionality to the PRE. Subsequently, Yau et al. [67] refined this approach by separating the document encryption and keyword encryption processes, thereby improving performance. Unfortunately, the search functionality of the PRE-KS only supports the cloud server to search for keywords that are encrypted under the same public key. Inspired by existing schemes, Li et al. [68] integrated proxy re-encryption with PKEET, resulting in a proxy re-encryption with equality test (PRE-ET) scheme. It not only has a ciphertext conversion function but also supports ciphertext matching, enabling secure data deduplication and cross-user decryption.

Specifically, in a typical PRE-ET [69] workflow, Alice encrypts her data and uploads it to the cloud server. When the cloud detects that Bob has uploaded a duplicate ciphertext through equality testing, it can use a re-encryption key that is generated from Alice's private key sk_A and Bob's public key pk_B to re-encrypt the stored ciphertext. The transformed ciphertext becomes decryptable by Bob's private key sk_B, allowing him to access the data securely without learning Alice's private key. The combination of data sharing with equality testing, particularly in PRE-ET schemes [70]–[73], has now become a practical solution for secure storage and efficient retrieval. Chen et al. [70] employed smart contracts to achieve reliable matching results and used PRE-ET to achieve efficient data sharing and privacy protection. Li et al. [72] further combined PRE and proposed a data sharing scheme that supports temporary delegation and hierarchical data retrieval, which improved the flexibility of ciphertext retrieval and the fine-grained sharing of decryption permissions. Nevertheless, current PRE-ET schemes still suffer from unresolved issues.

- High computational and communication overhead. Most schemes rely on bilinear pairings to perform re-encryption and equality testing, resulting in a heavy computational burden and high latency, especially unsuitable for IoMT with constrained resources.
- Lack of subversion resistance. Existing constructions rarely consider algorithm substitution attacks, where a malicious proxy or eavesdropper may embed subverted algorithms to leak sensitive information.

Therefore, we should consider efficiency, generality, and subversion resistance characteristics to ensure both security and practicality in multi-user data sharing systems.

C. Mass Surveillance Defense Methods

Various defense mechanisms have been proposed to address the risk of data leakage and subversion, as shown in Table I. One such approach involves deploying multiple devices from independent sources [50], ensuring that an attacker must compromise all components simultaneously to access sensitive information. Although effective, this strategy faces challenges such as high deployment costs and limited scalability. Bellare et al. [51] have demonstrated that all randomized encryption schemes are vulnerable to algorithm substitution attacks without additional protective measures. As a result, deterministic or unique ciphertext/signature schemes [52], [53] are considered resistant to subversion, as they ensure that each plaintext maps to at most one ciphertext under an untainted decryption algorithm. However, these schemes sacrifice flexibility and randomness, reducing their security in practical applications. Given the limitations of these techniques, subsequent research has focused on developing more general defenses to resist mass surveillance.

(1) Decomposition-amalgamation approach. This separates cryptographic algorithms into two distinct components: deterministic and randomized parts. By doing so, each part can be independently verified and recombined. It effectively limits the attack surface and constrains potential malicious behavior by isolating different functions. Meanwhile, the watchdog model introduces an external verifier that monitors communication between entities. The verifier is responsible for detecting anomalies or unexpected behaviors caused by subverted internal components. It enhances the system's ability to identify malicious actions without relying solely on internal components for integrity checks. However, it may introduce challenges in efficiency and scalability when deployed in complex systems.

(2) Self-guarding cryptographic protocol. Proposed by Fischlin et al. [57], the approach assumes that there is a trusted initialization phase (security anchor), after which the protocol relies on a provable structure that ensures security even when internal components become compromised. This model enables the protocol to self-defend against internal

TABLE II: Summary of functional comparison with other schemes.

Schemes	Techniques	Multi-hop	Non-interactive	Security	Autonomous path	Subversion resistance	Data matching
[4], [14]	PRE	×	✓	CPA	×	×	×
[15], [16]	PRE	×	✓	CCA	×	×	×
[46]	PRE	✓	✓	CPA	×	×	×
[48]	PRE, Homomorphic	✓	✓	CPA	×	×	×
[49]	AP-PRE	✓	✓	CPA	✓	×	×
[23]	PBRE	✓	✓	CCA	×	×	×
[12], [26], [45], [47]	PRE	✓	✓	CCA	×	×	×
[32]	AP-PRE, Searchable	✓	×	CPA	✓	×	✓
[70], [72]	PRE-ET	×	✓	CCA, CPA	×	×	✓
[68]	PRE-ET	✓	✓	CPA	×	×	✓
[69]	PRE-ET	×	✓	CCA	×	×	✓
[74]	ABE, CRF	×	×	CPA	×	✓	×
[75]	PKEET, CRF	×	×	CPA	×	✓	✓
[76], [77]	PRE, CRF	×	✓	CPA	×	✓	×
Ours	AP-PRE-ET, CRF	✓	✓	CPA	✓	✓	✓

threats without requiring external intervention. However, it may be less flexible and harder to implement in environments with highly dynamic or distributed components.

(3) Cryptographic reverse firewall. Compared with the above methods, the cryptographic reverse firewall (CRF) introduced by Mironov et al. [56] offers a distinctive advantage: it is stackable. Multiple CRFs can be deployed concurrently, and confidentiality is preserved as long as at least one CRF behaves honestly. Because adversaries cannot determine which CRF is effective at any given time, the overall defense achieves greater robustness. In this sense, CRFs provide a general and practical framework for resisting algorithm substitution attacks. Unlike traditional firewalls that focus on external intrusions, a CRF is designed to counter internal exfiltration. It serves as an intermediary layer between internal entities and the external environment, rerandomizing and sanitizing all input and output messages so that compromised components cannot leak sensitive information. A sound CRF satisfies three requirements: functionality preservation, security preservation, and exfiltration resistance. With these properties and its modular deployment model, CRFs protect confidential data even when parts of the execution environment are untrusted.

Subsequent research has further expanded the CRF framework. Dodis et al. [78] explored a CRF-based model for secure message transmission under chosen-ciphertext attacks (CCA), while Chen et al. [79] developed an extensible smooth projective hash function (SPHF) to create a generalized CRF framework. Further studies have integrated CRFs into various cryptographic settings. For example, Ma et al. [74] combined CRFs with a CPA-secure online/offline attribute-based encryption scheme, and Hong et al. [80] developed a key-policy ABE with multiple authorities supporting non-monotonic access structures. Zhou et al. [81], [82] proposed CRF-enhanced identity-based and certificateless encryption/signature schemes, while more recent work by Zhou et al. [76] identified ASAs in proxy re-encryption. Elhabob et al. [75] introduced a CRF-based public-key encryption scheme with an equality test, and Eltayieb et al. [77] designed an efficient certificateless proxy re-encryption scheme resistant to subversion. Therefore, CRFs remain attractive because of their simplicity, practicality, and generality. By leveraging rerandomizable encryption and key malleability, they can ensure that confidential information remains protected as long as at least one CRF functions honestly. These characteristics make CRFs an effective defense against both exfiltration and subversion attacks.

However, existing CRF-based encryption techniques remain incompatible with PRE-ET schemes. In addition, prior research on PRE-ET has not adequately addressed the weakening of trust in multi-hop delegation or the deduplication of outsourced data in cloud storage. As summarized in Table II, current PRE variants provide only partial support for the desired functionalities. In contrast, our SRAP-PRET scheme is the first to support all target properties simultaneously, effectively addressing previous limitations and achieving enhanced security properties guarantees while maintaining efficiency.

III. Preliminaries

This section outlines the background concepts and notations used in the proposed scheme, with the relevant symbols summarized in Table III.

A. Bilinear Map and Assumption

Definition 1 (Bilinear Map). Suppose that there is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of prime order q, and let g be a randomly chosen generator of $\mathbb{G}_1$. Then, the pairing e is required to satisfy the following properties [83]:

(1) Bilinearity: $e(g_1^u, g_2^v) = e(g_1, g_2)^{uv}$ holds for all $u, v \in Z_q^*$ and $g_1, g_2 \in \mathbb{G}_1$.
(2) Non-degeneracy: There exists g such that $e(g, g) \neq 1$.
(3) Computability: $e(g_1, g_2)$ can be computed efficiently for any $g_1, g_2 \in \mathbb{G}_1$.

Definition 2 (Decisional Bilinear Diffie-Hellman (DBDH) Problem). Given $(g, g^a, g^b, g^c, T) \in \mathbb{G}_1^4 \times \mathbb{G}_2, a, b, c \xleftarrow{\$} \mathbb{Z}_q^*$, decide whether $T = e(g, g)^{abc}$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are

cyclic multiplicative groups of the same prime order q, and g is a generator of $\mathbb{G}_1$. Let $\kappa = |q|$ denote the security parameter. The *DBDH Problem* is said to hold in $(\mathbb{G}_1, \mathbb{G}_2)$ if, for any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, its distinguishing advantage is negligible concerning the security parameter κ

$$\big|\ \Pr\big[\mathcal{A}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 1\big] - \Pr\big[\mathcal{A}(g, g^a, g^b, g^c, T) = 1\big]\ \big|.$$

B. Algorithm Substitution Attack

Definition 3 (ASA). An algorithm substitution attack (ASA) [51] occurs when an adversary replaces the legitimate encryption algorithm $\mathscr{E}$ in a scheme $\Sigma = (K, \mathscr{E}, \mathscr{D})$ with a modified version $\tilde{\mathscr{E}}_{\tilde{K}}$ that uses an attacker-chosen key $\tilde{K}$. The adversary has two main goals:

- Obtain confidential information, such as the original plaintext or the secret key, by analyzing only the ciphertexts produced by $\tilde{\mathscr{E}}_{\tilde{K}}$.
- Ensure that the modified ciphertexts are indistinguishable from those produced by the original algorithm so that the change is not detected.

The attack is considered successful if both goals are achieved.

ASAs can undermine the privacy, authenticity, or both aspects of an encryption scheme, depending on how the attack is designed. They can target symmetric encryption, public key encryption, or other cryptographic protocols such as key exchange. The main danger is that the adversary can insert a hidden information-leak mechanism into the encryption process, allowing them to recover sensitive data without affecting the execution correctness of the algorithm.

C. Cryptographic Reverse Firewall

The cryptographic reverse firewall (CRF) [56], [79], denoted $\mathcal{W}$, is a stateful algorithm acting as an intermediary between a party $\mathcal{P}$ and its environment, modifying both outgoing and incoming messages. Typically, a stateful composite party denoted as $\mathcal{W} \circ \mathcal{P}$ is defined to replace $\mathcal{P}$, where $\mathcal{W}$ is commonly referred to as a CRF specifically designed for $\mathcal{P}$. When CRF is deployed in party $\mathcal{P} = (\text{receive}, \text{next}, \text{output})$ with three cryptographic algorithms, the composed party $\mathcal{W} \circ \mathcal{P}$ that satisfies

$$\begin{aligned}\mathcal{W} \circ \mathcal{P} := (&\text{receive}_{\mathcal{W}\circ\mathcal{P}}(\sigma, m) = \text{receive}_{\mathcal{P}}(\sigma, \mathcal{W}(m)),\\ &\text{next}_{\mathcal{W}\circ\mathcal{P}}(\sigma) = \mathcal{W}(\text{next}_{\mathcal{P}}(\sigma)),\\ &\text{output}_{\mathcal{W}\circ\mathcal{P}}(\sigma) = \text{output}_{\mathcal{P}}(\sigma)),\end{aligned}$$

where σ represents an initial public parameter.

Definition 4 (Reverse Firewall (RF)). Let Π be a cryptographic scheme satisfying functionality $\mathcal{F}$ and security requirements $\mathcal{S}$, a CRF $\mathcal{W}$ deployed on the $\mathcal{P}$ should fulfill the following three requirements:

(1) Functionally-maintaining RFs. For $k \geq 1$, $\mathcal{W}$ maintains $\mathcal{F}$ in Π for $\mathcal{P}$ if $\mathcal{W}^k \circ \mathcal{P}$ maintains functionality requirements $\mathcal{F}$ for $\mathcal{P}$ in Π. For $k \geq 2$, let $\mathcal{W}^k \circ \mathcal{P} = \mathcal{W} \circ (\mathcal{W}^{k-1} \circ \mathcal{P})$. Specifically, when Π, $\mathcal{F}$ and $\mathcal{P}$ are clear, we simply say that $\mathcal{W}$ preserves functionally.

(2) Weakly security-preserving RFs. For a cryptographic scheme Π, $\mathcal{W}$ weakly preserves $\mathcal{S}$ for $\mathcal{P}$ in Π against $\mathcal{F}$-maintaining adversaries[1] if the scheme $\Pi_{\mathcal{P}\Rightarrow\mathcal{W}\circ\overline{\mathcal{P}}}$ fulfills $\mathcal{S}$ for any corrupted entity $\overline{\mathcal{P}}$ of $\mathcal{P}$. Specifically, when Π, $\mathcal{F}$ and $\mathcal{P}$ are clear, we simply say that $\mathcal{W}$ weakly preserves security.

(3) Weakly exfiltration-resistant RFs. For a cryptographic scheme Π, $\mathcal{W}$ is weakly exfiltration-resistant against any $\mathcal{F}$-maintaining adversaries if any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ has only a negligible advantage in winning `Game-LEAK` (See section III-E for details).

TABLE III: Summary of notations.

Symbol	Description
GP	Global parameters
Pa_i	The autonomous delegation path created by user i
i_j	The j-th delegatee on tha path Pa_i
l_i	The number of delegatees in Pa_i
u	The total number of users
(pk_i, sk_i)	Public and private key pair
$rk^i_{j\to j+1}$	The re-encryption key from delegatee i_j to delegatee i_{j+1} in Pa_i
m	Message
w	Keyword corresponding to the message
c_0^i	First-layer/Non re-encrypted ciphertext
$c_j^{i\prime}$	Re-encrypted ciphertext

D. Algorithm Definition

Definition 5 (Subversion-resistant autonomous path proxy re-encryption with equality test (SRAP-PRET)). The syntax of our proposed SRAP-PRET scheme, and the set of algorithms are presented as follows.

- GlobalSetup: Given the security parameter κ, this algorithm yields GP as the global parameters.
- KeyGen: This algorithm takes a user's identity i as an input, outputs the public and private key pair (pk_i, sk_i).
- $\mathcal{W}_{\mathsf{PKG}}$.KeyGen: This algorithm is operated by $\mathcal{W}_{\mathsf{PKG}}$, which takes GP, pk_i, and sk_i as input, outputs the re-randomized public and private key pair (pk_i', sk_i') for the registered user.
- Encrypt: This algorithm inputs GP, the sensitive EHR m, a keyword w corresponding to EHR, and pk_i', outputs the first-layer ciphertext c_0^i.
- $\mathcal{W}_{\mathsf{CL}}$.Encrypt: This algorithm operated by $\mathcal{W}_{\mathsf{CL}}$, which inputs GP and the first-layer ciphertext c_0^i, outputs the re-randomized ciphertext $c_0^{i\prime}$.
- TrapdoorGen: This algorithm inputs GP, sk_i', and $c_0^{i\prime}$, outputs the delegatee i's trapdoor td_i. Similarly, when the user i_j executes the algorithm, it produces a trapdoor td_{i_j}.

[1] $\overline{\mathcal{P}}$ is the adversarial, functionality-maintaining version of $\mathcal{P}$, and $\Pi_{\mathcal{P}\Rightarrow\mathcal{W}\circ\overline{\mathcal{P}}}$ indicates using $\mathcal{W} \circ \overline{\mathcal{P}}$ to replace $\mathcal{P}$.

- $\mathcal{W}_{\mathsf{CL}}$.TrapdoorGen: This algorithm operated by $\mathcal{W}_{\mathsf{CL}}$, which inputs GP and td_{i_j}, outputs the re-randomized trapdoor td'_{i_j}.
- Test: On input GP, two ciphertexts $c_0^{i'}$, $c_j^{i'}$, and two trapdoors td'_{i_0} and td'_{i_j}, this algorithm outputs 0 or 1. Both ciphertexts can be identified as coming from the same plaintext or not.
- CreatPath: This algorithm inputs GP, pk'_i, and outputs a delegation path Pa_i. The autonomous delegation path $Pa_i = (pk'_{i_0} = pk'_i, pk'_{i_1}, ..., pk'_{i_{l_i}})$ signifies a sequence of l_i different public keys in order, containing l_i delegatees. In Pa_i, pk'_{i_1} represents the first delegatee, pk'_{i_2} represents the second delegatee, and so on.
- RKeyGen: This algorithm inputs GP, sk'_i, and Pa_i, outputs l_i re-encryption keys $\{rk^i_{j-1\to j}\}_{j=1}^{l_i}$. Such as $rk^i_{j\to j+1}$ represents the transfer on the i's delegation path Pa_i from delegatee i_j to i_{j+1}.
- $\mathcal{W}_{\mathsf{CL}}$.RKeyGen: This algorithm operated by $\mathcal{W}_{\mathsf{CL}}$, which inputs GP and $\{rk^i_{j-1\to j}\}_{j=1}^{l_i}$, outputs re-randomized re-encryption keys $\{rk^{i'}_{j-1\to j}\}_{j=1}^{l_i}$.
- ReEncrypt: This algorithm inputs GP, a delegation path Pa_i, $rk^{i'}_{j\to j+1}$, and the matching ciphertext. It begins by verifying if $(pk'_{i_j}, pk'_{i_{j+1}})$ is contained within Pa_i. If yes, it re-encrypts the matching ciphertext and outputs the re-encrypted ciphertext. Otherwise, it outputs "$\perp$".
- Decrypt: The decryption algorithm inputs GP, the corresponding ciphertext and private key, outputs m or an error symbol $\perp$.

E. Security Model

In this section, we formalize two security games for the SRAP-PRET scheme, namely `Game-CPA` and `Game-LEAK`. `Game-CPA` defines a unidirectional, multi-hop CPA security model under adaptive corruption. In this game, the adversary may adaptively issue private-key and re-encryption queries, but must comply with the path rule, which permanently forbids oracle calls that would otherwise reveal the challenge message. `Game-LEAK` captures algorithm substitution attacks, allowing the adversary to replace any algorithm except the RF operation and thereby attempt to compromise the system. ASA is characterized by the adversary's ability to manipulate algorithms in a covert manner, potentially leading to the leakage of sensitive information. While the classical AP-PRE scheme [49] establishes semantic security under the chosen plaintext attack model (IND-CPA), our SRAP-PRET scheme advances this security model by formally capturing resistance to exfiltration via algorithm substitution attacks, even under insider adversaries capable of eavesdropping.

Definition 6 (`Game-CPA`). An SRAP-PRET is IND-CPA-CRF secure if the winning advantage of any PPT adversary $\mathcal{A}$ in the following `Game-CPA` is negligible.

Setup: $\mathcal{B}$ executes GlobalSetup algorithm to obtain GP and transmits to $\mathcal{A}$.

Find stage: During this stage, $\mathcal{A}$ is permitted to issue a number of queries bounded by a polynomial, $\mathcal{B}$ responds to $\mathcal{A}$'s queries as follows:

- *PKQuery oracle* $\mathcal{O}_{pk}$: Given GP and an identity index $i \in \{1, ..., u\}$, $\mathcal{B}$ runs the KeyGen algorithm to obtain (pk_i, sk_i). Then, $\mathcal{B}$ produces the rerandomizable key pair (pk'_i, sk'_i) by running $\mathcal{W}_{\mathsf{PKG}}$.KeyGen algorithm, and sends pk'_i to $\mathcal{A}$. Finally, $\mathcal{B}$ records the rerandomizable key pairs in the table Γ_{pk}.
- *SKQuery oracle* $\mathcal{O}_{sk}$: Given pk'_i, $i \in \{1, ..., u\}$, $\mathcal{B}$ searches the public key pk'_i in the table Γ_{pk} and provides $\mathcal{A}$ with the corresponding private key sk'_i.
- *Encryption oracle* $\mathcal{O}_{en}$: When $\mathcal{A}$ inputs an identity i, $\mathcal{B}$ first computes the ciphertext c_0^i and executes $\mathcal{W}_{\mathsf{CL}}$.Encrypt algorithm to obtain the rerandomizable ciphertext $c_0^{i'}$, then sends $c_0^{i'}$ to $\mathcal{A}$.
- *TrapdoorQuery Oracle* $\mathcal{O}_{td}$: $\mathcal{B}$ generates a rerandomizable trapdoor td'_{i_j} by running TrapdoorGen and $\mathcal{W}_{\mathsf{CL}}$.TrapdoorGen algorithms, then transmits it to $\mathcal{A}$.
- *Path creation oracle* $\mathcal{O}_{cp}$: If $\mathcal{A}$ has queried $\mathcal{O}_{cp}(i, Pa'_i)$, $\mathcal{B}$ responds with "$\perp$". Otherwise, $\mathcal{B}$ creates the delegation path $Pa_i = (pk'_{i_0} = pk'_i, pk'_{i_1}, ..., pk'_{i_{l_i}})$ by executing CreatPath algorithm if $\mathcal{A}$ is querying $\mathcal{O}_{cp}^i$ w.r.t. i for the first time. Then $\mathcal{B}$ gets the rerandomizable re-encryption key $\{rk^{i'}_{j-1\to j}\}_{j=1}^{l_i}$ for the path Pa_i by running RKeyGen and $\mathcal{W}_{\mathsf{CL}}$.RKeyGen algorithms. Last, $\mathcal{B}$ stores Pa_i in the table Γ_P, records $\{rk^{i'}_{j-1\to j}\}_{j=1}^{l_i}$ in the table Γ_{rk}, and responds to $\mathcal{A}$ with "Pa_i has been created".
- *RKQuery oracle* $\mathcal{O}_{rk}$: On input $(i, pk'_{i_j}, pk'_{i_{j+1}})$, $\mathcal{B}$ initially verifies whether the table Γ_P includes the user i's path. If yes, $\mathcal{B}$ retrieves the corresponding re-encryption key from Γ_{rk} and outputs it. Otherwise, $\mathcal{B}$ outputs "$\perp$".
- *Re-encryption oracle* $\mathcal{O}_{reen}$: Takes $(i, pk'_{i_j}, pk'_{i_{j+1}}, c_j^{i'})$ as input, $\mathcal{B}$ first verifies whether the user i's path $Pa_i = (..., pk'_{i_j}, pk'_{i_{j+1}}, ...)$ exists in the table Γ_P. If yes, $\mathcal{B}$ executes the ReEncrypt algorithm to compute the re-encrypted ciphertext $c_{j+1}^{i'}$, and sends it to $\mathcal{A}$. Otherwise, $\mathcal{B}$ outputs "$\perp$".

Challenge: $\mathcal{A}$ decides when **Find stage** ends, and generates two equal length messages m_0 and m_1, which it wants to be challenged. $\mathcal{B}$ randomly selects a bit $b \in \{0, 1\}$ and computes the ciphertext $c^* = Enc(GP, pk'_i, m_b)$ under pk'_i in Pa_i, then $\mathcal{B}$ runs $\mathcal{W}_{\mathsf{CL}}$.Encrypt to generate the rerandomizable ciphertext $c^{*\prime}$, and returns $c^{*\prime}$ to $\mathcal{A}$. Here, some restrictions on adversary $\mathcal{A}$ are as follows:

- $\mathcal{A}$ has not executed any private key generation queries on pk'_i or pk'^2_j.
- $\mathcal{A}$ must comply with the path rule concerning the path Pa_i and the initial ciphertext c_0^i.
- For any path $Pa_j = (pk_{j_0} = pk_j, pk_{j_1}, ..., pk_{j_{l_j}})$ generated by $\mathcal{A}$ for user j, the adversary is required to follow the path rule defined below.

Consider any sub-path of Pa_i, denoted as $Pa^i_{0\to k} = (pk_{i_0} = pk_i, pk_{i_1}, \ldots, pk_{i_k})$, where $1 \le k \le \ell_i$ and ℓ_i is the total length of the path. For each v such that $0 \le v < k$, suppose the adversary $\mathcal{A}$ issues either a re-encryption key query $\mathcal{O}_{rk}(i, pk_{i_v}, pk_{i_{v+1}})$ or a re-encryption operation query $\mathcal{O}_{reen}(i, pk_{i_v}, pk_{i_{v+1}}, c_v^i)$, and obtains a ci-

[2] pk'_j is the public key of the delegatee i_μ on the delegation path of the delegator i (marked as j).

phertext which can be decrypted to one of the challenge messages, i.e., $\mathsf{Dec}(par, c^i_{v+1}, sk_{i_{v+1}}) \in \{m_0, m_1\}$. In this case, the following restriction must hold for every step along the sub-path:

- The adversary must not request the private key for $pk_{i_{v+1}}$ via $\mathcal{O}_{sk}(pk_{i_{v+1}})$ at any point during this game.

Guess: $\mathcal{A}$ can initiate queries similar to those in the find stage, while following the restrictions imposed during the challenge stage. Ultimately, $\mathcal{A}$ guesses the challenge bit $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins this game.

We define the advantage that $\mathcal{A}$ wins `Game-CPA` as $Adv_{\mathcal{A}}^{\mathsf{IND\text{-}CPA\text{-}CRF}}(\kappa) = |Pr[b' = b] - 1/2|$.

Definition 7 (`Game-LEAK`). An SRAP-PRET is weak exfiltration resistant secure if any PPT adversary $\mathcal{A}$ has only a negligible advantage in the following `Game-LEAK`.

Tampering: The adversary $\mathcal{A}$ selects specific tampered algorithms, namely KeyGen^*, $\mathsf{Encrypt}^*$, $\mathsf{TrapdoorGen}^*$, and $\mathsf{RKeyGen}^*$, then sends them to challenger $\mathcal{B}$. Upon receiving these tampered algorithms, $\mathcal{B}$ substitutes the original algorithms with these tampered versions.

Initialization: $\mathcal{B}$ runs $\mathsf{GlobalSetup}$ algorithm, and uses KeyGen^* algorithm to generate a public/private key pair (pk_i, sk_i). Then $\mathcal{B}$ sends pk_i and GP to $\mathcal{A}$, and sk_i must be absolutely confidential.

Phase 1: Similar to the **Find stage** of `Game-CPA`, except $\mathcal{B}$ replaces $\mathsf{Encrypt}$, $\mathsf{TrapdoorGen}$, $\mathsf{RKeyGen}$ algorithms with the $\mathsf{Encrypt}^*$, $\mathsf{TrapdoorGen}^*$, and $\mathsf{RKeyGen}^*$ algorithms.

Challenge: $\mathcal{A}$ determines the end of phase 1 and produces two messages of equal length m_0 and m_1, to be used as a challenge. $\mathcal{B}$ randomly selects a bit $b \in \{0, 1\}$, and computes the challenged rerandomizable ciphertext or the challenged private key to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ again issues queries as in phase 1, subject to the same limitations as in the `Game-CPA` challenge stage.

Guess: Ultimately, $\mathcal{A}$ guesses the challenge bit $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins this game.

We define the advantage that $\mathcal{A}$ wins `Game-LEAK` as $Adv_{\mathcal{A}}^{\mathsf{exfiltration\text{-}resist}}(\kappa) = |Pr[b' = b] - 1/2|$.

IV. Problem Statement

This section presents the architectural design, threat model, and security objectives underlying the SRAP-PRET scheme.

A. System Architecture

As shown in Fig. 2, the system consists of five different parties: a private key generator (PKG), clients (including data owner DO and data consumer DC), a cloud proxy server (CPS), and the cryptographic reverse firewall (CRF) zone. Note that each CRF operates as an independent component located between internal entities and the outside world to enhance the security of the system. In the overall workflow, the PKG distributes keys to registered users, the DO uploads ciphertexts to the CPS, then CPS manages storage and provides much better fine-grained access control, and the authorized DC obtains shared data through the CPS after verification success. The specific workflow is as follows:

- **System Initialization:** The PKG executes the $\mathsf{GlobalSetup}$ and KeyGen algorithms to generate public-private key pairs (pk, sk), which are sanitized by $\mathcal{W}_{\mathsf{PKG}}$ to obtain (pk', sk'). The sanitized keys are then distributed to registered users.
- **Document Encryption:** The DO encrypts electronic health records (EHRs) with designated keywords and sends them to the client-side cryptographic reverse firewall $\mathcal{W}_{\mathsf{CL}}$, which sanitizes the original ciphertext by executing the $\mathcal{W}_{\mathsf{CL}}.\mathsf{Encrypt}$ algorithm and forwards it to CPS for storage.
- **Data Deduplication and Sharing:** During the deduplication phase, DO first generates a trapdoor for the ciphertext, which is sanitized by $\mathcal{W}_{\mathsf{CL}}$ and then uploaded to CPS. When a DC requests access to specific medical data, a corresponding trapdoor is generated, and the CPS is authorized to perform an equality test. Then, CPS verifies whether the DC belongs to the delegation path Pa_i predefined by the DO. If yes, CPS executes the Test algorithm to check if the ciphertext is derived from the same plaintext. When the test result is positive, CPS applies the $\mathsf{ReEncrypt}$ algorithm using the sanitized re-encryption keys associated with Pa_i to produce the re-encrypted ciphertext, which is then delivered to the DC. If the result is negative, DC will upload a new ciphertext as the data owner, enabling secure and efficient data sharing without duplicating storage.
- **Document Decryption:** Data owners can use their private key to decrypt the original ciphertext directly. The authorized data consumers can use their private key to decrypt the re-encrypted ciphertext and recover the shared data.

B. Threat Model

In this system, the PKG could potentially act as a malicious entity, manipulating generated system parameters and compromising users' private keys, effectively granting privileges to adversaries. Meanwhile, the honest-but-curious CPS provides ciphertext storage and conversion for users, and may attempt to analyze sensitive information related to their medical records.

Additionally, as Bellare et al. [18] demonstrated, all randomized cryptographic schemes cannot resist subversion attacks. The proxy re-encryption scheme is also inevitable. In such attacks, internal adversaries (e.g., an implementation device with a backdoor) would maliciously implement algorithms of the scheme, including the encryption and the re-encryption algorithms, to undermine the security. When users run algorithms in the subverted machine (which users cannot detect), embedded subliminal channels secretly leak some users' private information to adversaries through a public channel.

C. Design Goals

SRAP-PRET should fulfill the following design goals:

Subversion-resistant. In the face of internal threats from adversarial mass surveillance, this scheme should possess robustness and resistance to subversion. Intuitively, subversive attacks perform intentional subversion against randomized algorithms that expert detectors cannot detect, let alone users.

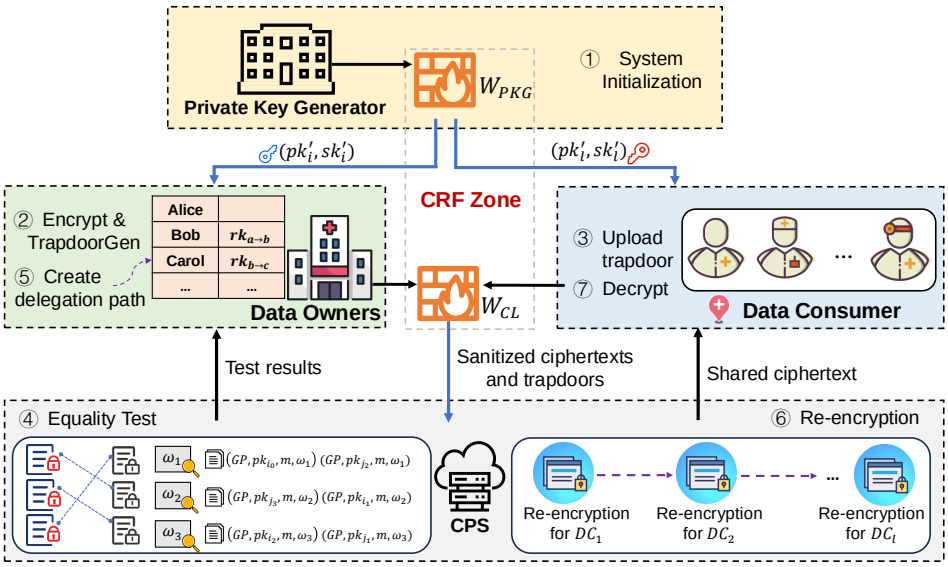


Fig. 2: System running flow.

Using deterministic algorithms instead of randomized algorithms is a simple method to achieve subversion resistance. However, this algorithm has a unique ciphertext property, which weakens the security of properly implemented schemes.

Delegation strictly follows the delegation path. This scheme should ensure that the ciphertext re-encrypted along the delegation path Pa_i set by the delegator cannot be converted to other paths. The re-encrypted ciphertext strictly adheres to the designated delegation path and remains indecipherable unless it conforms to Pa_i. Simultaneously, the first-level ciphertext cannot undergo conversion and integration into the autonomous path without preserving its decryptability.

Securely and efficiently. SRAP-PRET scheme should be secure against subversive attacks and chosen plaintext attacks (CPA). Meanwhile, CPS should provide efficient ciphertext conversion for authorized users but cannot decrypt matched ciphertexts. Furthermore, CPS is unable to acquire the data owner's private key or relevant keyword information.

V. Proposed SRAP-PRET

This section is organized into two parts: we first detail the construction of the SRAP-PRET, followed by a further discussion on its design rationale and potential extensions.

A. Construction of SRAP-PRET

We design a subversion-resistant PRE scheme with the autonomous path delegation function by deploying a CRF zone on the PKG and client sides, which are constructed as follows.

GlobalSetup: Upon receiving the security parameter κ, PKG yields the global parameters GP, which include: $\mathbb{G}_1$ and $\mathbb{G}_2$ represent two multiplicative groups with the same prime order q, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ represents a bilinear pair and two hash functions $H : \mathbb{G}_2 \to \mathbb{G}_1$, $H_2 : \mathbb{G}_1 \to Z_q^*$. Then, PKG selects $g, g_1 \in \mathbb{G}_1$, the message space $\mathcal{M} \in \mathbb{G}_2$. Finally, PKG outputs $GP = \{\mathbb{G}_1, \mathbb{G}_2, H, H_2, e, q, g, g_1\}$.

KeyGen: On input GP and a delegator's identity $i \in \{1, \cdots, u\}$, PKG selects $s_i \in Z_q^*$ randomly, and sets $pk_i = g^{s_i}$ as public key and $sk_i = s_i$ as private key. Then, PKG sends pk_i and sk_i to its CRF $\mathcal{W}_{\mathsf{PKG}}$.

$\mathcal{W}_{\mathsf{PKG}}$.KeyGen: After the $\mathcal{W}_{\mathsf{PKG}}$ receives the public and private key pair (pk_i, sk_i), it re-randomizes them and generates a fresh pair $(pk_i' = g^{\alpha s_i}, sk_i' = \alpha s_i)$, where $\alpha \in Z_q^*$. Finally, it sends (pk_i', sk_i') to the registered user.

Encrypt: Taking GP, pk_i', the sensitive medical data $m \in \mathcal{M}$, and a keyword $w \in Z_q^*$ as input, this algorithm randomly chooses $r \in Z_q^*$, and computes $c_{0_1}^i = g^r$, $c_{0_2}^i = m \cdot e(g_1, pk_{i_0}')^r$, $c_{0_3}^i = g^{H_2({pk_{i_0}'}^r)+w}$. Last, it outputs the first-layer ciphertext $c_0^i = (c_{0_1}^i, c_{0_2}^i, c_{0_3}^i)$.

$\mathcal{W}_{\mathsf{CL}}$.Encrypt: Given GP, pk_{i_0}' and c_0^i, $\mathcal{W}_{\mathsf{CL}}$ selects $\beta \in Z_q^*$ and produces a randomized ciphertext $c_{0_1}^{i'} = c_{0_1}^i \cdot g^\beta$, $c_{0_2}^{i'} = c_{0_2}^i \cdot e(g_1, pk_{i_0}')^\beta \cdot e(c_{0_1}^i \cdot g^\beta, {pk_{i_0}'}^\beta)$, $c_{0_3}^{i'} = g^{\beta H_2({pk_{i_0}'}^r)+w}$. After that, it outputs the re-randomized ciphertext $c_0^{i'} = (c_{0_1}^{i'}, c_{0_2}^{i'}, c_{0_3}^{i'})$.

TrapdoorGen: Given GP, sk_{i_0}' and c_0^i, the delegator generates a trapdoor $td_{i_0} = H_2(c_{0_1}^{i\ sk_{i_0}'})$ for $\mathcal{W}_{\mathsf{CL}}$. Similarly, a delegatee i_j executes this algorithm and outputs a trapdoor $td_{i_j} = H_2(c_{j_1}^{i\ sk_{i_j}'})$, where $j \in \{0, \cdots, l_i\}$.

$\mathcal{W}_{\mathsf{CL}}$.TrapdoorGen: Given GP and td_{i_j}, $\mathcal{W}_{\mathsf{CL}}$ generates the re-randomized trapdoor $td_{i_j}' = \beta H_2(c_{j_1}^{i\ sk_{i_j}'})$ and submits to CPS for equivalence testing.

Test: The CPS first checks $c_{0_3}^{i'} \cdot g^{-td_{i_0}'} \stackrel{?}{=} c_{j_3}^{i'} \cdot g^{-td_{i_j}'}$, it returns 1 if it holds, and 0 otherwise. 1 indicates that the ciphertext is successfully matched, then CPS runs the ReEncrypt algorithm to re-encrypt the matched ciphertext.

CreatPath: Given GP, pk_i' (denoted by pk_{i_0}' in our scheme), it outputs a delegation path $Pa_i = (pk_{i_0}' = pk_i', pk_{i_1}', \cdots, pk_{i_{l_i}}')$ which contains l_i delegatees.

RKeyGen: Given GP, sk_i', and a delegation Pa_i, to generate the corresponding re-encryption keys, this algorithm randomly selects $X_j \in \mathbb{G}_2$, $r_j \in Z_q^*$, where $j = 1, \cdots, l_i$.

(i) For $j = 1$, it computes $rk^i_{0\to1} = (rk^i_{(0\to1)_1}, rk^i_{(0\to1)_2}, rk^i_{(0\to1)_3}) = (g^{r_1}, X_1 \cdot e(g_1, pk'_{i_1})^{r_1}, H(X_1) \cdot g_1^{-sk'_i})$.

(ii) For $j = 2, \cdots, l_i$, computes $rk^i_{j-1\to j} = (rk^i_{(j-1\to j)_1}, rk^i_{(j-1\to j)_2}, rk^i_{(j-1\to j)_3}) = (g^{r_j}, X_j \cdot e(g_1, pk'_{i_j})^{r_j}, \frac{H(X_j)}{H(X_{j-1})})$.

Then, the re-encryption key $\{rk^i_{j-1\to j}\}^{l_i}_{j=1}$ is sent to the client side $\mathcal{W}_{\mathsf{CL}}$.

$\mathcal{W}_{\mathsf{CL}}$.RKeyGen: After receiving $\{rk^i_{j-1\to j}\}^{l_i}_{j=1}$, it generates the randomized re-encryption key $\{rk^{i'}_{j-1\to j}\}^{l_i}_{j=1}$.

(i) For $j = 1$, $rk^{i'}_{0\to1} = (rk^{i'}_{(0\to1)_1}, rk^{i'}_{(0\to1)_2}, rk^{i'}_{(0\to1)_3}) = (rk^i_{(0\to1)_1} \cdot g^\beta, rk^i_{(0\to1)_2} \cdot e(g_1, pk'_{i_1})^\beta, rk^i_{(0\to1)_3} \cdot pk_{i_1}'^{-\beta})$.

(ii) For $j = 2, \cdots, l_i$, $rk^{i'}_{j-1\to j} = (rk^{i'}_{(j-1\to j)_1}, rk^{i'}_{(j-1\to j)_2}, rk^{i'}_{(j-1\to j)_3}) = (rk^i_{(j-1\to j)_1} \cdot g^\beta, rk^i_{(j-1\to j)_2} \cdot e(g_1, pk'_{i_j})^\beta, rk^i_{(j-1\to j)_3})$.

ReEncrypt: Given GP, a delegation path $Pa_i = (pk'_{i_0} = pk'_i, pk'_{i_1}, \cdots, pk'_{i_{l_i}})$, $rk^{i'}_{j\to j+1}$ and $c^{i'}_j$. CPS performs the following algorithm to convert the ciphertext encrypted under the public key pk'_{i_j} to the ciphertext encrypted under the $pk'_{i_{j+1}}$:

(i) First, verify if $(pk'_{i_j}, pk'_{i_{j+1}}) \in Pa_i$ and output "$\perp$" if not.

(ii) Otherwise, if $j = 0$, CPS parses the ciphertext $c^{i'}_j = (c^{i'}_{j_1}, c^{i'}_{j_2}, c^{i'}_{j_3})$. In this case, the randomized re-encryption key is like $rk^{i'}_{(j\to j+1)} = rk^{i'}_{0\to1} = (rk^i_{(0\to1)_1} \cdot g^\beta, rk^i_{(0\to1)_2} \cdot e(g_1, pk'_{i_1})^\beta, rk^i_{(0\to1)_3} \cdot pk_{i_1}'^{-\beta})$. If $j \geq 1$, CPS parses $c^{i'}_j = (c^{i'}_{j_1}, c^{i'}_{j_2}, c^{i'}_{j_3}, c^{i'}_{j_4} = rk^{i'}_{(j-1\to j)_1}, c^{i'}_{j_5} = rk^{i'}_{(j-1\to j)_2})$, where the randomized re-encryption key is like $rk^{i'}_{(j\to j+1)} = (rk^{i'}_{(j\to j+1)_1}, rk^{i'}_{(j\to j+1)_2}, rk^{i'}_{(j\to j+1)_3})$. Finally, CPS computes $c^{i'}_{j+1} = (c^{i'}_{(j+1)_1}, c^{i'}_{(j+1)_2}, c^{i'}_{(j+1)_3}, c^{i'}_{(j+1)_4}, c^{i'}_{(j+1)_5}) = (c^{i'}_{j_1}, c^{i'}_{j_2} \cdot e(c^{i'}_{j_1}, rk^{i'}_{(j\to j+1)_3}), c^{i'}_{j_3}, rk^{i'}_{(j\to j+1)_1}, rk^{i'}_{(j\to j+1)_2})$ as the re-encrypted ciphertext, which can be decrypted by $sk'_{i_{j+1}}$ and sends $c^{i'}_{j+1}$ to the authorized delegatee.

Decrypt: On input GP, a ciphertext $c^{i'}_j$ and the private key sk'_{i_j}, this algorithm checks whether or not $c^{i'}_j$ is an original ciphertext or one that has been re-encrypted. Then, for these two types of ciphertext, execute the following corresponding decryption algorithms:

(i) For the original ciphertext that has not undergone re-encryption, c^i_j is first transmitted by the CPS to $\mathcal{W}_{\mathsf{CL}}$. Then, $\mathcal{W}_{\mathsf{CL}}$ computes $c^{i'}_j{}'' = (c^{i''}_{j_1}, c^{i''}_{j_2}, c^{i''}_{j_3}) = (c^{i'}_{j_1}, c^{i'}_{j_2} \cdot e(c^i_{0_1} \cdot g^\beta, pk_{i_j}'^{-\beta}), c^{i'}_{j_3})$ and sends $c^{i'}_j{}''$ to the data owner. Finally, DO can compute $m = \frac{c^{i''}_{j_2}}{e(c^{i''}_{j_1}, g_1^{sk'_{i_j}})}$ to recover.

(ii) In situations involving re-encrypted ciphertext, the authorized delegatee parses $c^{i'}_j = (c^{i'}_{j_1}, c^{i'}_{j_2}, c^{i'}_{j_3}, c^{i'}_{j_4}, c^{i'}_{j_5})$. First, computes $X_j = \frac{c^{i'}_{j_5}}{e(c^{i'}_{j_4}, g_1^{sk'_{i_j}})}$ with the public key pk'_{i_j}, then decrypts $m = \frac{c^{i'}_{j_2}}{e(c^{i'}_{j_1}, H(X_j))}$. Finally, the authorized delegatee can obtain the decryption result.

B. Further Discussion

In SRAP-PRET, the delegator is given a flexible and self-defined trusted delegation path, which enables precise control over data ownership during multi-hop ciphertext conversion, solving the problems of weakened trust and abuse of authority. However, extending the access link will inevitably increase the average complexity of encrypted data access, where high-priority delegatees usually limit the decryption task complexity of low-priority delegatees. A delegated branch path can be created to further optimize the decryption efficiency, and a branch token can be embedded in the ciphertext structure. The delegatee on the main path employs the token to establish a branch path. Therefore, it is attractive to introduce such a subversion-resistant PRE that supports branch functionality. In brief, we may nontrivially combine PRE with autonomous path delegation, branch link construction technology, and a malleable CRF built upon SPHF to realize a general framework for SRAPB-PRE.

VI. Security Analysis

We first discuss the SRAP-PRE scheme, which satisfies the following security requirements. Subsequently, we provide security proof for the scheme in this section.

A. Security Properties of SRAP-PRET

- **Confidentiality.** The confidentiality ensures that sensitive information in medical record reports cannot be accessed by unauthorized users. If there are security vulnerabilities or backdoors in the client's implementation devices, there will be a risk of information exfiltration; that is, the r randomly selected by the Encrypt algorithm will be leaked. We utilized $\mathcal{W}_{\mathsf{CL}}$ to encapsulate the ciphertext with a random value β, effectively mitigating such attacks.
- **Forgery resistance.** We assume an extreme case that CPS is a malicious proxy, thus it can forge a DO_i by calculating $c^{i^*}_0 = Enc(GP, pk_i, M^*)$ and $c^{i^*}_j = ReEnc(GP, rk^i_j, c^{i^*}_0)$, generate a forged re-encrypted ciphertext $c^{i^*}_j$, then send it to the recipient. In our scheme, a fake ciphertext forged by a malicious CPS cannot be decrypted by the recipient. Because $\mathcal{W}_{\mathsf{CL}}$ uses β to re-randomize the re-encryption keys, the connection between the ciphertexts and their re-encrypted ciphertexts is blurred to achieve ciphertext unlinkability. Therefore, deducing the ciphertext after the corresponding re-encryption key conversion from pk_i is impossible.
- **Delegation path is strictly followed.** When a data owner i establishes a path $Pa_i = (pk'_{i_0} = pk'_i, pk'_{i_1}, ..., pk'_{i_{l_i}})$ based on personal preferences beforehand, the patient's ciphertext access right must strictly follow the path Pa_i for delegation. This implies that the delegatee j's re-encrypted ciphertext in $Pa_i (j \neq i)$ cannot be transformed and inserted into any other path, including Pa_j which was created by j or i. Likewise, the original ciphertext produced by i cannot be transformed and inserted into another path.

B. Security Proof for SRAP-PRET

Theorem 1. The proposed SRAP-PRET scheme maintains functionality, weak security preservation, and weak resistance to exfiltration. More precisely, weakly security-preserving means that when the computer executing the cryptographic algorithm is tampered with, CRF can provide the same security as the original protocol. Weakly exfiltration-resistant means that for an adversary with ASA ability, CRF can prevent internal information leakage and the normal functioning of the scheme will not be affected.

Proof. We demonstrate that SRAP-PRET satisfies the following three properties:

(1) Functionally Maintaining. Only the data owner can decrypt the re-randomized ciphertext $c_j^{i\prime\prime} = (c_{j_1}^{i\prime\prime}, c_{j_2}^{i\prime\prime}, c_{j_3}^{i\prime\prime})$, when the ciphertext is non re-encrypted ciphertext.

$$\begin{aligned}\frac{c_{j_2}^{i\prime\prime}}{e(c_{j_1}^{i\prime\prime}, g_1^{sk'_{i_j}})} &= \frac{c_{j_2}^{i\prime\prime} \cdot e(c_{0_1}^{i} \cdot g^{\beta}, pk'^{\,-\beta}_{i_0})}{e(c_{j_1}^{i'}, g_1^{sk'_{i_j}})} \\ &= \frac{m \cdot e(g_1, pk'_{i_0})^{r} \cdot e(g_1, pk'_{i_0})^{\beta}}{e(c_{j_1}^{i'}, g_1^{sk'_{i_j}})} \\ &= \frac{m \cdot e(g_1^{r+\beta}, g^{\alpha s_i})}{e(g^{r+\beta}, g_1^{\alpha s_i})} = m.\end{aligned}$$

In the case of a re-encrypted ciphertext, the re-randomize ciphertext $c_j^{i'} = (c_{j_1}^{i'}, c_{j_2}^{i'}, c_{j_3}^{i'}, c_{j_4}^{i'}, c_{j_5}^{i'})$ can be decrypted through the following procedure:

(i) For $j = 1$, the re-randomize ciphertext is like $(c_{1_1}^{i'}, c_{1_2}^{i'}, c_{1_3}^{i'}, c_{1_4}^{i'}, c_{1_5}^{i'})$, where

$$\begin{aligned}c_{1_1}^{i'} &= g^{r+\beta}, \\ c_{1_2}^{i'} &= c_{1_2}^{i'} \cdot e(c_{1_1}^{i'}, rk_{(0\to1)_3}^{i'}) = m \cdot e(g_1, pk'_{i_0})^{r+\beta} \\ &\quad \cdot e(g^{r+\beta}, pk'_{i_0})^{\beta} \cdot e(g^{r+\beta}, H(X_1) \cdot g_1^{-sk'_{i_0}} \cdot pk'^{-\beta}_{i_0}) \\ &= m \cdot e(g_1, pk'_{i_0})^{r+\beta} \cdot e(g^{r+\beta}, H(X_1) g^{-sk'_i}) \\ &= m \cdot e(g^{r+\beta}, H(X_1)), \\ c_{1_4}^{i'} &= rk_{(0\to1)_1}^{i'} = g^{r_1+\beta}, \\ c_{1_5}^{i'} &= rk_{(0\to1)_2}^{i'} = X_1 \cdot e(g_1, pk'_{i_1})^{r_1} \cdot e(g_1, pk'_{i_1})^{\beta}.\end{aligned}$$

The recipient receives $c_j^{i'}$, first computes

$$\begin{aligned}\frac{c_{1_5}^{i'}}{e(g_1^{sk'_{i_1}}, c_{1_4}^{i'})} &= \frac{X_1 \cdot e(g_1, pk'_{i_1})^{r_1} \cdot e(g_1, pk'_{i_1})^{\beta}}{e(g_1^{\alpha s_i}, g^{r_1+\beta})} \\ &= \frac{X_1 \cdot e(g_1^{r_1+\beta}, g^{\alpha s_i})}{e(g_1^{\alpha s_i}, g^{r_1+\beta})} \\ &= X_1.\end{aligned}$$

Then computes

$$m = \frac{c_{1_2}^{i'}}{e(c_{1_1}^{i'}, H(X_1))} = \frac{m \cdot e(g^{r+\beta}, H(X_1))}{e(g^{r+\beta}, H(X_1))} = m.$$

(ii) For $j > 1$, the re-randomize ciphertext is like $c_j^{i'} = (c_{j_1}^{i'}, c_{j_2}^{i'}, c_{j_3}^{i'}, c_{j_4}^{i'}, c_{j_5}^{i'})$, where

$$\begin{aligned}c_{j_1}^{i'} &= g^{r+\beta}, \\ c_{j_2}^{i'} &= c_{j_2}^{i'} \cdot e(c_{j_1}^{i'}, rk_{(j-1\to j)_3}^{i'}) \\ &= m \cdot e(c_{j_1}^{i'}, H(X_{j-1})) \cdot e(c_{j_1}^{i'}, \frac{H(X_j)}{H(X_{j-1})}) \\ &= m \cdot e(g^{r+\beta}, H(X_j)), \\ c_{j_4}^{i'} &= rk_{(j-1\to j)_1}^{i'} = g^{r_j+\beta}, \\ c_{j_5}^{i'} &= rk_{(j-1\to j)_2}^{i'} = X_j \cdot e(g_1, pk'_{i_j})^{r_j} \cdot e(g_1, pk'_{i_j})^{\beta}.\end{aligned}$$

The process for the recipient with pk'_{i_j} to recover the data is similar to the above, so we omit it. Therefore, deploying CRFs keeps the solution's functionality intact.

(2) Weakly Security Preservation. The IND-CPA-CRF security of our scheme is substantiated through the tempered algorithms KeyGen*, Encrypt*, TrapdoorGen*, and RKeyGen*, as evidenced by the `Game-CPA` detailed in section III-E and the underlying AP-PRE scheme are indistinguishable. Therefore, we consider the following series of games to prove their indistinguishability:

`Game 0`: This game is identical to `Game-CPA`, which is an IND-CPA-CRF game that was introduced in section III-E.

`Game 1`: This game is similar to `Game 0`, with the distinction that in the IND-CPA game, the users' keys are generated by the KeyGen algorithm during the **Find stage**, not the KeyGen* and $\mathcal{W}_{\mathsf{PKG}}$.KeyGen algorithms in `Game 0`.

`Game 2`: This game is similar to `Game 1`, with the distinction that the trapdoor is generated by TrapdoorGen algorithm in the IND-CPA security game during the **Find stage**, not the TrapdoorGen* and $\mathcal{W}_{\mathsf{CL}}$.TrapdoorGen algorithms in `Game 1`.

`Game 3`: This game is similar to `Game 2`, with the distinction that the re-encryption keys are generated by RKeyGen algorithm in the IND-CPA security game during the **Find stage**, not the RKeyGen* algorithm and $\mathcal{W}_{\mathsf{CL}}$.RKeyGen algorithm in `Game 2`.

`Game 4`: This game is similar to `Game 3`, with the distinction that Encrypt algorithm generates the ciphertext in the IND-CPA security game during the **Find stage**, not the Encrypt* algorithm and $\mathcal{W}_{\mathsf{CL}}$.Encrypt algorithm in `Game 3`.

`Game 5`: This game is similar to `Game 4`, with the distinction that Encrypt algorithm generates the challenge ciphertext in the IND-CPA security game during the **Find stage**, not the Encrypt* and $\mathcal{W}_{\mathsf{CL}}$.Encrypt algorithms in the challenge stage. Essentially, `Game 5` is equivalent to the conventional standard security challenge defined in the AP-PRE [49] framework.

Following, we will prove the indistinguishability between any adjacent games in `Game 0`∼`Game 5`.

`Game 0` $\approx$ `Game 1`. CRF will sanitize potentially biased key pair (pk_i, sk_i) to produce a re-randomized key pair (pk'_i, sk'_i) in the tampered algorithm KeyGen*. Due to the key malleability property, the re-randomized key pair is a consistent random number, which is consistent with the KeyGen algorithm. Hence, this captures the indistinguishability of `Game 0` and `Game 1`.

TABLE IV: Computational and communication overhead.

Scheme	Computational overhead							Communication overhead		
	ReKey	Encrypt	ReEnc	Trapdoor	Test	$\text{Decrypt}_{\text{or}}$	$\text{Decrypt}_{\text{re}}$	Ciphertext	ReKey	Trapdoor
PBRE [23]	$T_p + 7T_e$	$T_p + 6T_e$	$8T_p + T_e$	–	–	$2T_p + 6T_e$	$6T_p + 6T_e$	$4\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$	$5\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$	–
AP-PRE [49]	$l_i(T_p + 3T_e)$	$T_p + 2T_e$	l_iT_p	–	–	$T_p + T_e$	$2T_p + T_e$	$\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	–
AP-PECKS [32]	$l_i(3T_e)$	$T_p + 6T_e$	l_iT_p	$5T_e$	$4T_p + 5T_e$	–	–	$5\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$\lvert\mathbb{G}_1\rvert + \lvert Z_q\rvert$
PRE-ET [68]	$3T_e$	$T_p + 8T_e$	$2T_e$	T_e	$4T_e$	T_e	$4T_e$	$4\lvert\mathbb{G}_1\rvert + \lvert Z_q\rvert$	$2\lvert\mathbb{G}_1\rvert + \lvert Z_q\rvert$	$\lvert\mathbb{G}_1\rvert$
CLPRE-CRF [77]	$l_i(T_p + 6T_e)$	$T_p + 4T_e$	$l_i(T_p + 5T_e)$	–	–	$T_p + 2T_e$	$T_p + T_e$	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$3\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	–
Ours	$l_i(2T_p + 5T_e)$	$3T_p + 7T_e$	l_iT_p	0	$2T_e$	$T_p + T_e$	$2T_p + T_e$	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$\lvert Z_q\rvert$

`Game 1` $\approx$ `Game 2`. For the tampered algorithm $\mathsf{TrapdoorGen}^*$, $\mathcal{W}_{\mathsf{CL}}$ will sanitize potentially biased trapdoor td_{i_j} to yield re-randomized trapdoors $td'_{i_j}, j \in \{0, 1, ..., l_i\}$. Owing to the element re-randomizability characteristic, the trapdoor is uniformly random, which is consistent with the trapdoor generated by $\mathsf{TrapdoorGen}$ algorithm. Hence, this captures the indistinguishability of `Game 1` and `Game 2`.

`Game 2` $\approx$ `Game 3`. For the tampered algorithm $\mathsf{RKeyGen}^*$, $\mathcal{W}_{\mathsf{CL}}$ will sanitize potentially biased re-encryption key $\{rk^i_{j-1\to j}\}_{j=1}^{l_i}$ to yield re-randomized re-encryption key $\{rk^{i'}_{j-1\to j}\}_{j=1}^{l_i}$. Owing to the key malleability characteristic, the re-encryption key is a consistent random number as that generated by $\mathsf{RKeyGen}$ algorithm. Hence, this captures the indistinguishability of `Game 2` and `Game 3`.

`Game 3` $\approx$ `Game 4`. For the tampered algorithm $\mathsf{Encrypt}^*$ that may be executed by $\mathcal{W}_{\mathsf{CL}}$. The $\mathcal{W}_{\mathsf{CL}}$.Encrypt algorithm generates a uniformly random re-randomized ciphertext $c_0^{i'}$ because of the element re-randomizability property, which is the same as the Encrypt algorithm in the AP-PRE scheme. Hence, this captures the indistinguishability of `Game 3` and `Game 4`.

`Game 4` $\approx$ `Game 5`. For the tampered algorithm $\mathsf{Encrypt}^*$ that may be executed by $\mathcal{W}_{\mathsf{CL}}$. The $\mathcal{W}_{\mathsf{CL}}$.Encrypt algorithm generates a uniformly random re-randomized challenge ciphertext $c^{*\prime}$ because of the element re-randomizability property, which is the same as the Encrypt algorithm in the AP-PRE scheme. Hence, this captures the indistinguishability of `Game 4` and `Game 5`.

Therefore, it can be inferred that `Game 0` and `Game 5` are indistinguishable, the proposed SRAP-PRET scheme achieves IND-CPA-CRF security, as the underlying AP-PRE [49] achieves IND-CPA security. Besides, we can also say that CRFs maintain weak security-preserving.

(3) Weakly Exfiltration Resistance. The aforementioned proof shows that the indistinguishability between `Game 0` and `Game 5` allows the adversary to merely win the `Game-LEAK` with a negligible advantage. Therefore, CRFs deployed on the PKG and client have weak exfiltration-resistance property. □

VII. Performance Evaluation

A. Theoretical Analysis

To provide a comprehensive evaluation of the efficiency of SRAP-PRET, we analyze its computational, communication, and storage overhead in comparison with representative schemes [23], [32], [49], [68], [77]. The most expensive operations are bilinear pairings T_p and exponentiations T_e, as other cryptographic operations incur negligible delay. Let $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, $|Z_q|$ be the length of the elements of $\mathbb{G}_1$, $\mathbb{G}_2$, Z_q, respectively. Table IV summarizes the computational and communication overhead of each algorithm. The $\mathsf{GlobalSetup}$ algorithm, executed only once, is excluded as it does not affect users' encryption or search experience.

TABLE V: Comparison of role storage overheads.

Scheme	Data owner	Cloud proxy server
PBRE [23]	$(2\kappa + 3)\lvert\mathbb{G}_1\rvert$	$(5l_i + 4)\lvert\mathbb{G}_1\rvert + (l_i + 1)\lvert\mathbb{G}_2\rvert + (l_i + 1)\lvert Z_q\rvert$
AP-PRE [49]	$3\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$	$(3l_i + 1)\lvert\mathbb{G}_1\rvert + (l_i + 1)\lvert\mathbb{G}_2\rvert$
AP-PECKS [32]	$(n + 6)\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + 2\lvert Z_q\rvert$	$(n + 2l_i + 8)\lvert\mathbb{G}_1\rvert + (l_i + 3)\lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$
PRE-ET [68]	$4\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + 2\lvert Z_q\rvert$	$(2l_i + 3)\lvert\mathbb{G}_1\rvert + 2\lvert\mathbb{G}_2\rvert + l_i\lvert Z_q\rvert$
CLPRE-CRF [77]	$4\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + \lvert Z_q\rvert$	$(3l_i + 2)\lvert\mathbb{G}_1\rvert + (l_i + 1)\lvert\mathbb{G}_2\rvert$
Ours	$3\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + 2\lvert Z_q\rvert$	$(3l_i + 2)\lvert\mathbb{G}_1\rvert + (l_i + 1)\lvert\mathbb{G}_2\rvert$

κ denotes the security parameter; l_i denotes the number of delegatees; n denotes the number of messages/keywords.

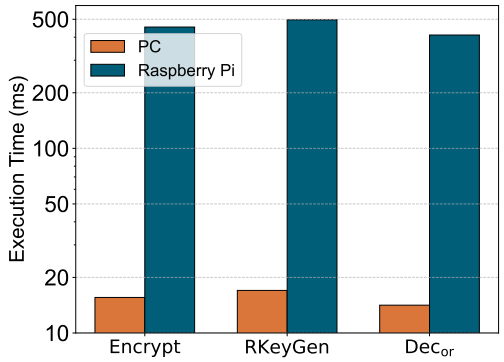
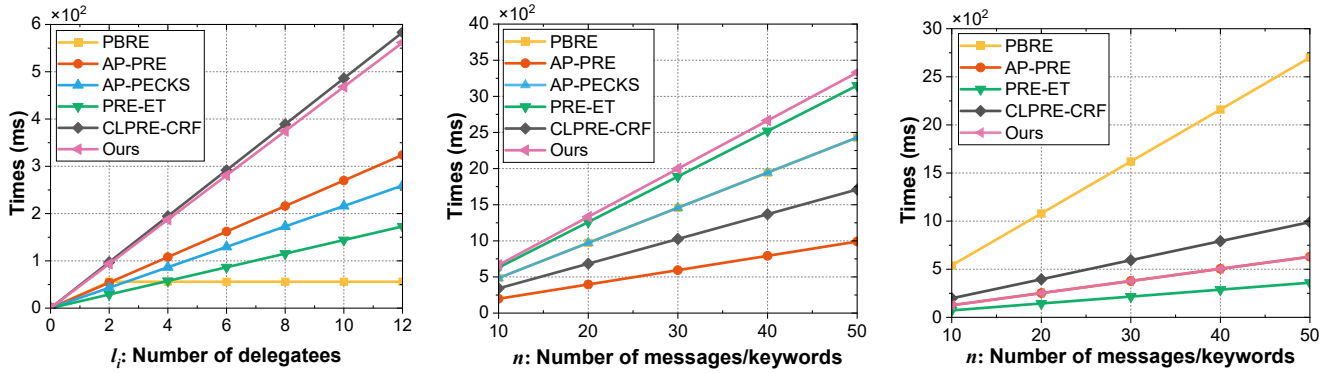


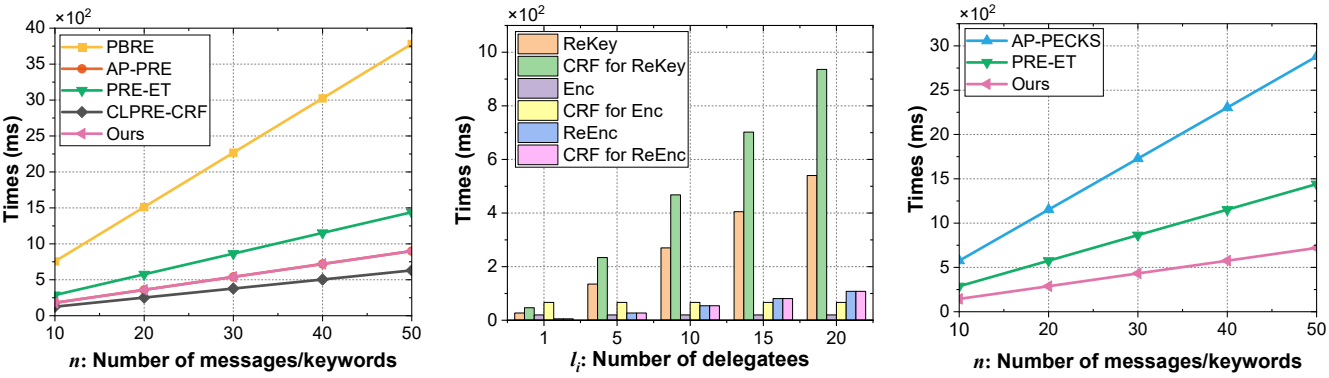
Fig. 3: Execution time comparison between PC and Raspberry Pi.

Data owner (DO). The major costs come from the $\mathsf{RKeyGen}$ and $\mathsf{Encrypt}$ algorithms. Both grow linearly with the number of delegatees l_i and the number of messages/keywords n. Compared with these schemes, SRAP-PRET introduces slightly higher costs due to the "sanitization" enforced by the CRF. Specifically, the computational cost for DO in the absence of CRF deployment comprises two distinct components: (i) generation of the re-encryption key rk^i that takes $l_i(T_p+3T_e)$; (ii) encryption of the transmission of the message that requires $T_p + 2T_e$. Thus, the original computational delay of DO is $(l_i + 1)T_p + (3l_i + 2)T_e$. Deploying CRF between DO and the external environment can achieve unbiased randomness of rk^i and prevent the malicious proxy from forging confidential information, so its additional computational cost is $(l_i + 2)T_p + (2l_i + 5)T_e$. It should be emphasized that these overheads are incurred by the CRFs, which ensure the



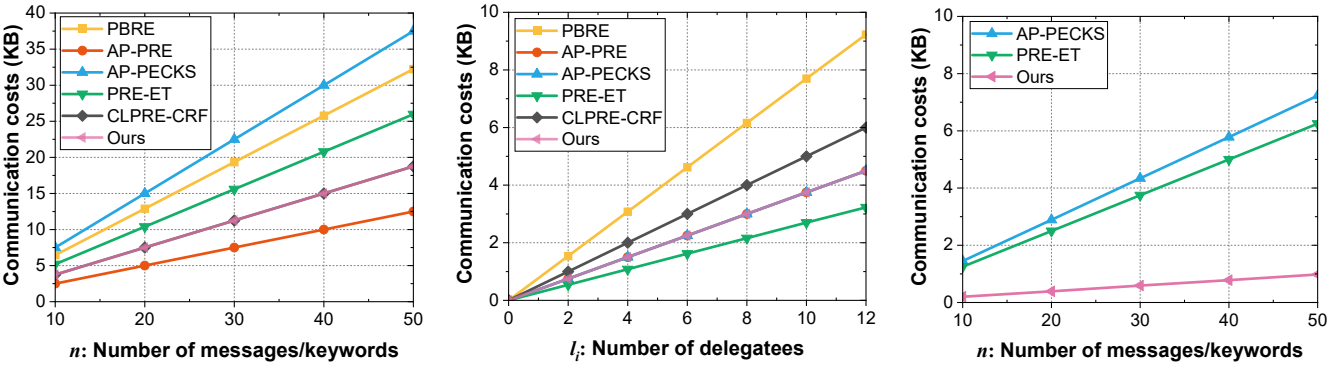
(a) RKeyGen algorithm. (b) Enc algorithm. (c) $\mathsf{Dec}_{\mathsf{or}}$ algorithm.

Fig. 4: Comparison in the computational overhead of DO.



(a) $\mathsf{Dec}_{\mathsf{re}}$ algorithm. (b) Computational delay. (c) Test algorithm.

Fig. 5: Comparison in the computational overhead of DC & CPS.



(a) Ciphertext. (b) ReKey (c) Trapdoor

Fig. 6: Comparison in the communication overhead of DO.

scheme's security against subversive attacks.

Cloud proxy server (CPS). The major costs come mainly from re-encryption and deduplication operations. Specifically, CPS needs to complete the following tasks: (i) check whether the submitted ciphertext of interest has been stored in the cloud, and obtain the test results by performing equality tests on the submitted trapdoor and the corresponding ciphertext (each test involves two exponentiation) to achieve deduplication; (ii) re-encryption on the matching ciphertexts according to the order of the shared paths Pa_i, where the cost scales with the number of delegatees but remains unaffected by CRF deployment.

Data consumer (DC). The major costs come mainly from the TrapdoorGen and Decrypt algorithms. Our scheme does not need pairing T_p or exponentiation T_e operations. In contrast, the remaining protocols listed in Table IV demand multiple pairing or exponentiation operations, resulting in a substantial energy drain on resource-constrained devices. The procedure of decrypting computational complexity is quantified as $2T_p + T_e$, which scales linearly with the number

of associated messages/keywords n. Even at $n = 50$, the decryption completes within about one second, demonstrating the practicality of the scheme.

Finally, we also analyze the storage overhead for different roles, as shown in Table V. The storage cost of the DO's side in SRAP-PRET is only marginally higher than that of AP-PRE [49], yet significantly smaller than AP-PECKS [32], and comparable to PRE-ET [68] and CLPRE-CRF [77]. The CPS-side storage overhead in all schemes varies with l_i (the number of delegatees) because CPS is usually used to perform complex re-encryption operations. In summary, the storage cost of our scheme remains comparable to or even smaller than existing schemes, while achieving strong security guarantees.

B. Experimental Simulation

The implementation and performance evaluation of SRAP-PRET, along with the schemes presented in [49], [23], [68], [77], and [32], were carried out using the Java programming language and the JPBC cryptographic library[3]. Traditionally, in medical scenarios, the user refers to individuals operating mobile devices or PCs. In our evaluation, we emulate a PC user with a machine equipped with an Intel Core i7-8565U processor (1.80 GHz), and a mobile user with a Raspberry Pi 3B+ powered by a BCM2837B0 core (1.4 GHz). We utilize Type A pairing on the curve $y^2 = x^3 + x$ over the field $E(F_q)$, where $r\text{Bits} = 160$ and $q\text{Bits} = 512$. On both devices, we measured the execution time of the algorithms performed at the data owner's side, namely Encrypt, RKeyGen, and $\mathsf{Dec_{or}}$. The results are presented in Fig. 3. RKeyGen incurs the highest computational cost, whereas Encrypt and $\mathsf{Dec_{or}}$ introduce comparatively modest overheads. Notably, even on the resource-constrained Raspberry Pi, all operations can be completed within a few hundred milliseconds. It demonstrates that SRAP-PRET remains computationally feasible for practical deployment in medical IoT environments where mobile or embedded devices act as data owners.

To further evaluate efficiency, we next focus on the computational overhead of the data owner (DO) across different schemes. As analyzed in the previous section, the sanitization of CRF requires a certain amount of computational cost, as shown in Fig. 4(a) and Fig. 4(b). Although SRAP-PRET has higher computational overhead in these two phases, it still maintains high efficiency in the Decrypt phase, as shown in Fig. 4(c). This demonstrates that the additional security mechanisms in SRAP-PRET do not impose prohibitive burdens on the user. Fig. 5(b) shows the latency of the RKeyGen algorithm, Encrypt algorithm, and ReEnc algorithm after deploying the CRF zone. This overhead is acceptable for both users and CPS. The comparison is provided in Fig. 5(c), which shows that the matching test latency increases with the number of requests linearly, and SRAP-PRET has the lowest cost.

We divide the communication cost into the ciphertext generation phase, re-encryption key generation phase, and trapdoor generation phase. For ciphertext transmission, the experimental results are illustrated in Fig. 6(a), where $|c_j^{i\prime}|$ and n are fixed to 20 KB and 50, respectively. SRAP-PRET bolsters security against subversion attacks has been enhanced at the expense of slight communication costs. For re-encryption keys, SRAP-PRET still maintains the same communication overhead as in [49] and [32], as illustrated in Fig. 6(b). Importantly, in the trapdoor phase, SRAP-PRET achieves the lowest cost among all compared schemes, reflecting its suitability for resource-constrained devices that frequently issue search queries. The experimental results confirm that SRAP-PRET strikes a favorable balance between efficiency and security, ensuring practicality in large-scale deployments.

[3]http://gas.dia.unisa.it/projects/jpbc/.

VIII. Conclusion

This work introduces a subversion-resistant proxy re-encryption scheme, which realizes the autonomous path delegation function and supports ciphertext matching. Delegators possess the prerogative to selectively assign search and decryption permissions for their medical data to the delegatees of their trusts, prioritizing from highest to lowest. In SRAP-PRET, we use CRF to sanitize transmitted messages, trapdoors, and re-encryption keys, achieving ciphertext unlinkability and thwarting subversion attacks. Subsequently, we formally prove that SRAP-PRET resists information leakage attacks and achieves IND-CPA security as well as unforgeability. Theoretical and experimental results indicate that SRAP-PRET is effective in computational and communication costs.

References

[1] Raza Nowrozy, Khandakar Ahmed, ASM Kayes, Hua Wang, and Timothy R McIntosh. Privacy preservation of electronic health records in the modern era: A systematic survey. *ACM Computing Surveys*, 56(8):1–37, 2024.

[2] S. Ahmed, M. Alam, S. Afrin, S. Rafa, N. Rafa, and A. Gandomi. Insights into internet of medical things (IoMT): Data fusion, security issues and potential solutions. *Information Fusion*, 102:102060, 2024.

[3] IBM Security. Medical devices are vital, but vulnerable. Website, 2023. https://www.ibm.com/downloads/cas/EQVODWLY.

[4] M. Green and G. Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007. Proceedings 5*, pages 288–306. Springer, 2007.

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, 2006.

[6] J. Liu, Q. Zhong, R. Sun, X. Du, and M. Guizani. A secure and efficient medical data sharing protocol for cloud-assisted WBAN. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.

[7] L. Fang, C. Yin, L. Zhou, Y. Li, C. Su, and J. Xia. A physiological and behavioral feature authentication scheme for medical cloud based on fuzzy-rough core vector machine. *Information Sciences*, 507:143–160, 2020.

[8] J. Hao, W. Tang, C. Huang, J. Liu, H. Wang, and M. Xian. Secure data sharing with flexible user access privilege update in cloud-assisted IoMT. *IEEE Transactions on Emerging Topics in Computing*, 10(2):933–947, 2021.

[9] J. Zhao, K. Zhang, J. Gong, and H. Qian. Lavida: Large-universe, verifiable, and dynamic fine-grained access control for e-health cloud. *IEEE Transactions on Information Forensics and Security*, 19:2732–2745, 2024.

[10] M. Ali, A. Hosseingholizadeh, and X. Liu. Data inspection and access control for 5G edge computing-enabled internet of medical things. *IEEE Transactions on Network Science and Engineering*, 11(5):4120–4133, 2024.

[11] F. Zhang, Z. Liang, C. Zuo, J. Shao, J. Ning, J. Sun, J. Liu, and Y. Bao. hpress: A hardware-enhanced proxy re-encryption scheme using secure enclave. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(6):1144–1157, 2020.

[12] S. Yao, R. Dayot, I.-H. Ra, L. Xu, Z. Mei, and J. Shi. An identity-based proxy re-encryption scheme with single-hop conditional delegation and multi-hop ciphertext evolution for secure cloud data sharing. *IEEE Transactions on Information Forensics and Security*, 18:3833–3848, 2023.

[13] Y. Zhou, S. Liu, and S. Han. Multi-hop fine-grained proxy re-encryption. In *IACR International Conference on Public-Key Cryptography*, pages 161–192. Springer, 2024.

[14] G. Ateniese, K. Benson, and S. Hohenberger. Key-private proxy re-encryption. In *Topics in Cryptology-CT-RSA 2009: The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings*, pages 279–294. Springer, 2009.

[15] L. Fang, W. Susilo, C. Ge, and J. Wang. Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search. *Theoretical Computer Science*, 462:39–58, 2012.

[16] S. Prasad and B. Purushothama. CCA secure and efficient proxy re-encryption scheme without bilinear pairing. *Journal of Information Security and Applications*, 58:102703, 2021.

[17] Q. Tang and M. Yung. Cliptography: Post-snowden cryptography. In *CCS 2017*, pages 2615–2616, 2017.

[18] M. Bellare, J. Jaeger, and D. Kane. Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In *CCS 2015*, pages 1431–1440, 2015.

[19] P. Prajapati and P. Shah. A review on secure data deduplication: Cloud storage security issue. *Journal of King Saud University-Computer and Information Sciences*, 34(7):3996–4007, 2022.

[20] April IDC. *The digital universe decade–are you ready?* Digital Universe Study, 2010.

[21] G. Yang, C. Tan, Q. Huang, and D. Wong. Probabilistic public key encryption with equality test. In *10th Cryptographers' Track at the RSA Conference, CT-RSA 2010 San Francisco, CA, USA, March 1-5, 2010.*, pages 119–131. Springer, 2010.

[22] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *International conference on the theory and applications of cryptographic techniques*, pages 127–144. Springer, 1998.

[23] M. Sun, C. Ge, L. Fang, and J. Wang. A proxy broadcast re-encryption for cloud data sharing. *Multimedia Tools and Applications*, 77:10455–10469, 2018.

[24] Z. Wang. Leakage resilient ID-based proxy re-encryption scheme for access control in fog computing. *Future Generation Computer Systems*, 87:679–685, 2018.

[25] H. Guo, Z. Zhang, J. Xu, N. An, and X. Lan. Accountable proxy re-encryption for secure data sharing. *IEEE Trans. Dependable Secure Comput.*, 18(1):145–159, 2018.

[26] H. Xiong, Y. Wang, W. Li, and C.-M. Chen. Flexible, efficient, and secure access delegation in cloud computing. *ACM Trans. Management Information Systems (TMIS)*, 10(1):1–20, 2019.

[27] C. Ge, Z. Liu, J. Xia, and L. Fang. Revocable identity-based broadcast proxy re-encryption for data sharing in clouds. *IEEE Transactions on Dependable and Secure Computing*, 18(3):1214–1226, 2019.

[28] H. Guo, Z. Zhang, J. Xu, and M. Xia. Generic traceable proxy re-encryption and accountable extension in consensus network. In *ESORICS 2019, Luxembourg, September 23-27, 2019, Part I 24*, pages 234–256. Springer, 2019.

[29] N. Eltayieb, L. Sun, K. Wang, and F. 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*, pages 293–307. Springer, 2019.

[30] J. Shen, H. Yang, P. Vijayakumar, and N. Kumar. A privacy-preserving and untraceable group data sharing scheme in cloud computing. *IEEE Trans. Dependable Secure Comput.*, 19(4):2198–2210, 2021.

[31] Y. Zhou, Z. Cao, X. Dong, and J. Zhou. BLDSS: A blockchain-based lightweight searchable data sharing scheme in vehicular social networks. *IEEE Internet of Things Journal*, 10(9):7974–7992, 2023.

[32] Q. Wang, C. Lai, R. Lu, and D. Zheng. Searchable encryption with autonomous path delegation function and its application in healthcare cloud. *IEEE Trans. on Cloud Comput.*, 11(1):879–896, 2023.

[33] Q. Tang. Type-based proxy re-encryption and its construction. In *International conference on cryptology in india*, pages 130–144. Springer, 2008.

[34] J. Weng, R. Deng, X. Ding, C.-K. Chu, and J. Lai. Conditional proxy re-encryption secure against chosen-ciphertext attack. In *Proceedings of the 4th international symposium on information, computer, and communications security*, pages 322–332, 2009.

[35] C.-K. Chu, J. Weng, S. Chow, J. Zhou, and R. Deng. Conditional proxy broadcast re-encryption. In *Australasian conference on information security and privacy*, pages 327–342. Springer, 2009.

[36] L. Fang, J. Wang, C. Ge, and Y. Ren. Fuzzy conditional proxy re-encryption. *Science China Information Sciences*, 56(5):1–13, 2013.

[37] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin. Conditional identity-based broadcast proxy re-encryption and its application to cloud email. *IEEE Transactions on Computers*, 65(1):66–79, 2015.

[38] S. Luo, J. Hu, and Z. Chen. Ciphertext policy attribute-based proxy re-encryption. In *International Conference on Information and Communications Security*, pages 401–415. Springer, 2010.

[39] H. Yin and L. Zhang. Security analysis and improvement of an anonymous attribute-based proxy re-encryption. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pages 344–352. Springer, 2017.

[40] H Ma, R Zhang, G Yang, Z Song, K He, and Y Xiao. Efficient fine-grained data sharing mechanism for electronic medical record systems with mobile devices. *IEEE Transactions on Dependable and Secure Computing*, 17(5):1026–1038, 2018.

[41] L. Xue, Y. Yu, Y. Li, M. Au, X. Du, and B. Yang. Efficient attribute-based encryption with attribute revocation for assured data deletion. *Information Sciences*, 479:640–650, 2019.

[42] K. Vohra and M. Dave. Securing fog and cloud communication using attribute based access control and re-encryption. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 307–312. IEEE, 2018.

[43] S. Maiti, S. Misra, and A. Mondal. ABP: Attribute-based broadcast proxy re-encryption with coalitional game theory. *IEEE Systems Journal*, 18(1):85–95, 2024.

[44] E. Kirshanova. Proxy re-encryption from lattices. In *International Workshop on Public Key Cryptography*, pages 77–94. Springer, 2014.

[45] H. Wang, Z. Cao, and L. Wang. Multi-use and unidirectional identity-based proxy re-encryption schemes. *Information Sciences*, 180(20):4042–4059, 2010.

[46] S. Luo, Q. Shen, and Z. Chen. Fully secure unidirectional identity-based proxy re-encryption. In *International Conference on Information Security and Cryptology*, pages 109–126. Springer, 2011.

[47] H.-Y. Lin. Secure content distribution using multi-hop proxy re-encryption. *Wireless Personal Communications*, 82:1449–1459, 2015.

[48] Z. Li, C. Ma, and D. Wang. Towards multi-hop homomorphic identity-based proxy re-encryption via branching program. *IEEE Access*, 5:16214–16228, 2017.

[49] Z. Cao, H. Wang, and Y. Zhao. AP-PRE: Autonomous path proxy re-encryption and its applications. *IEEE Trans. Dependable Secure Comput.*, 16(5):833–842, 2019.

[50] G. Li, J. Liu, and Z. Zhang. An overview on cryptography against mass surveillance. *Journal of Cryptologic Research*, 6(3):269–282, 2019.

[51] M. Bellare, K. Paterson, and P. Rogaway. Security of symmetric encryption against mass surveillance. In *CRYPTO 2014*, pages 1–19, Berlin, Heidelberg, 2014.

[52] M. Bellare and V. Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 627–656. Springer, 2015.

[53] S. Chow, A. Russell, Q. Tang, M. Yung, Y. Zhao, and H.-S. Zhou. Let a non-barking watchdog bite: Cliptographic signatures with an offline watchdog. In *IACR International Workshop on Public Key Cryptography*, pages 221–251. Springer, 2019.

[54] A. Young and M. Yung. Kleptography: Using cryptography against cryptography. In *EUROCRYPT'97: International Conference on the Theory and Application of Cryptographic Techniques Konstanz, Germany, May 11-15, 1997 Proceedings 16*, pages 62–74. Springer, 1997.

[55] A. Russell, Q. Tang, M. Yung, and H.-S. Zhou. Generic semantic security against a kleptographic adversary. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 907–922, 2017.

[56] I. Mironov and D. Stephens. Cryptographic reverse firewalls. In *Advances in Cryptology-EUROCRYPT 2015*, pages 657–686, Berlin, Heidelberg, 2015. Springer.

[57] M. Fischlin and S. Mazaheri. Self-guarding cryptographic protocols against algorithm substitution attacks. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 76–90. IEEE, 2018.

[58] R. Elhabob, M. Taha, H. Xiong, M. Khan, S. Kumari, and P. Chaudhary. Pairing-free certificateless public key encryption with equality test for internet of vehicles. *Computers and Electrical Engineering*, 116:109140, 2024.

[59] W. Li, W. Susilo, C. Xia, L. Huang, F. Guo, and T. Wang. Secure data integrity check based on verified public key encryption with equality test for multi-cloud storage. *IEEE Transactions on Dependable and Secure Computing*, 21(6):5359–5373, 2024.

[60] Z. Li, J. Lai, J. Han, L. Chen, and X. Yang. Forward secure equality test for secure data sharing in healthcare systems. *IEEE Internet of Things Journal*, 2025.

[61] J. Chen, S. Zeng, S. Han, J. Yin, and P. Chen. Pkest: Public-key encryption with similarity test for medical consortia cloud computing. *IEEE Transactions on Cloud Computing*, 2025.

[62] A. Hassan, Y. Wang, R. Elhabob, N. Eltayieb, and F. Li. An efficient certificateless public key encryption scheme with authorized equality test in healthcare environments. *Journal of Systems Architecture*, 109:101776, 2020.

[63] R. Elhabob, Y. Zhao, I. Sella, and H. Xiong. An efficient certificateless public key cryptography with authorized equality test in IIoT. *Journal of Ambient Intelligence and Humanized Computing*, 11(3):1065–1083, 2020.

[64] Y. Xu, M. Wang, H. Zhong, J. Cui, L. Liu, and V. Franqueira. Verifiable public key encryption scheme with equality test in 5G networks. *IEEE Access*, 5:12702–12713, 2017.

[65] K. Zhang, B. Hu, J. Ning, J. Gong, and H. Qian. Pattern hiding and authorized searchable encryption for data sharing in cloud storage. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[66] J. Shao, Z. Cao, X. Liang, and H. Lin. Proxy re-encryption with keyword search. *Information Sciences*, 180(13):2576–2587, 2010.

[67] W.-C. Yau, R. Phan, S.-H. Heng, and B.-M. Goi. Proxy re-encryption with keyword search: New definitions and algorithms. In *Security Technology, Disaster Recovery and Business Continuity*, pages 149–160, Berlin, Heidelberg, 2010. Springer.

[68] W. Li, C. Jin, S. Kumari, H. Xiong, and S. Kumar. Proxy re-encryption with equality test for secure data sharing in Internet of Things-based healthcare systems. *Transactions on Emerging Telecommunications Technologies*, 33(10):e3986, 2022.

[69] G. Han, L. Li, B. Qin, and D. Zheng. Pairing-free proxy re-encryption scheme with equality test for data security of iot. *Journal of King Saud University-Computer and Information Sciences*, 36(6):102105, 2024.

[70] B. Chen, D. He, N. Kumar, H. Wang, and K.-K. Choo. A blockchain-based proxy re-encryption with equality test for vehicular communication systems. *IEEE Trans. Network Science and Engineering*, 8(3):2048–2059, 2020.

[71] C.-C. Yang, R. Tso, Z.-Y. Liu, J.-C. Hsu, and Y.-F. Tseng. Improved proxy re-encryption scheme with equality test. In *2021 16th Asia Joint Conference on Information Security (AsiaJCIS)*, pages 37–44. IEEE, 2021.

[72] W. Li, C. Xia, C. Wang, and T. Wang. Secure and temporary access delegation with equality test for cloud-assisted IoV. *IEEE Trans. Intelligent Transportation Systems*, 23(11):20187–20201, 2022.

[73] W. Li, C. Xia, S. Yang, K. Wang, G. Huang, L. Huang, F. Guo, W. Susilo, and T. Wang. Fine-grained access control with privacy-preserving data retrieval for cloud-assisted iov. *IEEE Transactions on Vehicular Technology*, 2025.

[74] H. Ma, R. Zhang, G. Yang, Z. Song, S. Sun, and Y. Xiao. Concessive online/offline attribute based encryption with cryptographic reverse firewalls—secure and efficient fine-grained access control on corrupted machines. In *ESORICS 2018*, pages 507–526, Cham, 2018. Springer International Publishing.

[75] R. Elhabob, N. Eltayieb, H. Xiong, F. Khan, A. Bashir, S. Kumari, R. Alturki, and S. Kumar. Equality test public key encryption with cryptographic reverse firewalls for cloud-based E-commerce. *IEEE Trans. Consumer Electronics*, 2024.

[76] Y. Zhou, L. Zhao, Y. Jin, and F. Li. Backdoor-resistant identity-based proxy re-encryption for cloud-assisted wireless body area networks. *Information Sciences*, 604:80–96, 2022.

[77] N. Eltayieb, R. Elhabob, A. Abdelgader, Y. Liao, F. Li, and S. Zhou. Certificateless proxy re-encryption with cryptographic reverse firewalls for secure cloud data sharing. *Future Generation Computer Systems*, 162:107478, 2025.

[78] Y. Dodis, I. Mironov, and D. Stephens. Message transmission with reverse firewalls—secure communication on corrupted machines. In *CRYPTO 2016*, pages 341–372, Berlin, Heidelberg, 2016. Springer.

[79] R. Chen, Y. Mu, G. Yang, W. Susilo, F. Guo, and M. Zhang. Cryptographic reverse firewall via malleable smooth projective hash functions. In *ASIACRYPT 2016*, pages 844–876, Berlin, Heidelberg, 2016. Springer.

[80] B. Hong, J. Chen, K. Zhang, and H. Qian. Multi-authority non-monotonic KP-ABE with cryptographic reverse firewall. *IEEE Access*, 7:159002–159012, 2019.

[81] Y. Zhou, Y. Guan, Z. Zhang, and F. Li. Cryptographic reverse firewalls for identity-based encryption. In *International Conference on Frontiers in Cyber Security*, pages 36–52. Springer, 2019.

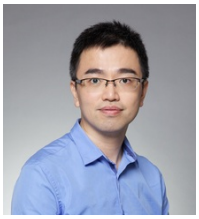
[82] Y. Zhou, J. Guo, and F. Li. Certificateless public key encryption with cryptographic reverse firewalls. *Journal of Systems Architecture*, 109:101754, 2020.

[83] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

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.

Lulu Wang received his Ph.D. in Software Engineering from East China Normal University, Shanghai, China, in 2025. He is currently a postdoc at The Hong Kong Polytechnic University. He was a visiting Ph.D. student at the SPRITZ Security and Privacy Research Group, University of Padua, Italy (2023), and a visiting scholar in the Information Systems Technology and Design Pillar at the Singapore University of Technology and Design (2024). His research interests are at the intersection of security, privacy, and machine learning.



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.

Zehui Xiong (Senior Member, IEEE) is currently a Full Professor with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, United Kingdom. Prior to that, he was with Singapore University of Technology and Design, and Nanyang Technological University (NTU). He received his Ph.D. degree from NTU and was a visiting scholar with Princeton University and University of Waterloo. Recognized as a Clarivate Highly Cited Researcher, he has published over 250 peer-reviewed research papers in leading journals, with numerous Best Paper Awards from international flagship conferences. Featured in Forbes Asia 30U30, he serves as the Editor for many leading journals and Chair for numerous international conferences. His honors include the IEEE Asia Pacific Outstanding Young Researcher Award, IEEE VTS Early Career Award, IEEE Early Career Award for Excellence in Scalable Computing, IEEE Technical Committee on Blockchain and Distributed Ledger Technologies Early Career Award, IEEE Internet Technical Committee Early Achievement Award, IEEE TCSVC Rising Star Award, IEEE TCI Rising Star Award, IEEE TCCLD Rising Star Award, IEEE ComSoc Outstanding Paper Award, IEEE Best Land Transport Paper Award, IEEE Asia Pacific Outstanding Paper Award, IEEE CSIM Technical Committee Best Journal Paper Award, IEEE SPCC Technical Committee Best Paper Award, and IEEE Big Data Best Influential Conference Paper Award.

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.