The ISC Int'l Journal of Information Security

ISeCure

November 2023, Volume 15, Number 3 (pp. 59–66)

http://www.isecure-journal.org

Presented at the ISCISC'2023 in Tehran, Iran.

An Efficient Scheme for Secure Medical Data Sharing in the Cloud [**]

Iman Jafarian [1], and Siavash Khorsandi [1,*]

[1] *Faculty of Computer Engineering, Amirkabir University of Technology, Tehran, Iran*

ARTICLE INFO.

Keywords:
Proxy Re-encryption, Elliptic Curve Cryptography, Secure Sharing, E-Health, Internet of Medical Things

Type:
Research Article

doi:
`10.22042/isecure.2023.182206`

ABSTRACT

The Internet of Things has significantly improved healthcare with its promise of transforming technological, social, and economic perspectives. Medical devices with wireless internet access enable remote monitoring of patients, and collectively, these increasingly smart and connected medical devices can provide unique and contemporary medical and health services envisioned to be deployed in a large-scale fashion. For this, medical data and records generally are collected, stored, and shared through open-air wireless networks and public cloud infrastructures, which poses severe challenges regarding the confidentiality of sensitive medical data while maintaining low service overhead and system complexity. This paper presents a novel scheme for secure cloud-assisted Internet of Medical Things connecting patients/smart medical devices to smart applications/medical service providers in a scalable one-to-many fashion to make novel medical services practical. The proposed scheme uses index-based searchable encryption for data screening without decryption. It uses a low-overhead proxy re-encryption scheme for secure data sharing through public clouds.

1 Introduction

In recent years, many remote and online services have been set to be provided through the Internet, among which remote medical services have received much attention [1]. In a remote medicine environment, various sensors on the body or in the surrounding environment of the patient send the patient's vital information to a medical center, and according to this data, the patient's health is monitored, and, if necessary, the required actions are performed by the medical center [2]. In addition, it is required that medical data related to patients is stored in a storage space in a safe manner where it can only be accessed and retrieved by the relevant/authorized personnel. Considering the abundance of data in the storage space and the importance of medical data's confidentiality and patient privacy, we need a cloud-assisted data-sharing scheme that allows doctors to search for the desired data before retrieving and decrypting the data records [3].

Medical data must be encrypted using symmetric or public key encryption algorithms to enable a safe

* Corresponding author.

**The ISCISC'2023 program committee effort is highly acknowledged for reviewing this paper.

Email addresses: `iman.j@aut.ac.ir`, `khorsandi@aut.ac.ir`

ISSN: 2008-2045 © 2023 ISC. All rights reserved.

exchange of medical data generated by the patients' wireless body area networks (WBANs). Nevertheless, flexible and scalable data sharing with numerous users is challenging with conventional encryption methods. Besides multiple key allocations overhead, there is an enormous extra communication burden associated with multiple data transfers to interested receivers, as identical medical data must be encrypted using various users' public/private keys and uploaded to the cloud to employ public key encryption. Also, the cloud stores duplicate ciphertexts of the same medical data, wasting storage space. Therefore, a safe and effective data exchange system in telemedicine applications is crucial for cloud-assisted WBANs [4].

This paper proposes a lightweight coupled one-way identifier-based proxy re-encryption protocol (LIPRE) based on elliptic curves to securely share patient health-related data with a semi-trusted publicly accessible cloud. Proxy re-encryption (PRE) is a method for converting encrypted data into a format that a specific receiver can decrypt it [5]. Patients encrypt their data using their public keys before outsourcing data to the cloud in proxy re-encryption. The cloud-resident semi-trusted proxy re-encrypts the data using the re-encryption key without knowing anything about the encrypted message, and the obtained encrypted data is stored on the cloud. In our proposed scheme, an index-based searchable encryption is applied to allow doctors to search among the data before opening the encrypted data. Due to the massive amount of data in the cloud storage space, we reduce the data screening overhead.

We provide a review of the related works, including proxy re-encryption schemes presented in recent years, in the following section. Section 3 describes the system model, including system architecture and adversary model. In Section 4, the proposed LIPRE protocol is presented. Section 5 includes the security analysis of the proposed scheme, and Section 6 provides a complexity analysis of the proposed scheme compared to the existing works. Finally, Section 7 concludes the paper.

2 Related Work

Wang [6] presented an identity-based proxy re-encryption (IBPRE) system with an ancillary input to resist a secret key token from the suggested channel in 2018. Nevertheless, as the number of subkeys increases, so does the length of the re-encryption key and the ciphertext. Xue *et al.* [7] presented an attribute-based PRE system (ABPRE) for fine-grained access control. The total number of characteristics in this system, which corresponds to the user's storage capacity, is directly connected to the size of the general parameters. To address the issue of the semi-trusted cloud in PRE, Qin *et al.* [8] developed a blockchain-based access control system. Data decoding is outsourced under this plan. For the Internet of Things, Su *et al.* [9] developed a PRE method based on trusted permission to ensure reliable updating of node authentication. To create a flexible definition of user identification, Liang *et al.* [10] used the idea of proxy re-encryption in attribute-based settings. Numerous ABPRE schemes have been created due to their efforts to expand access policy expression and improve the security model. Unfortunately, none of these ABPRE approaches considers user renunciation, which is crucial for systems that share data. By removing the proxies from the re-encryption key, Ge *et al.* [11] presented a proxy re-encryption strategy based on a revocable identifier to overcome the key revocation problem. However, a malicious proxy might tamper with the message and send it to the agent. The idea of Identity-Based Broadcast Encryption (IBBE), where the user's identity is considered the public key in Identity-Based Broadcast Encryption, was suggested by Sakai and Furukawa [12]. A broadcast proxy re-encryption technique by Chu *et al.* [13] allows a proxy to turn Alice's ciphertext into a collection of proxies.

By combining El-Gamal encryption with Schnorr's signature, Deng *et al.* [14] created a unique bidirectional PRE scheme safe against adaptively selected ciphertext attacks. This approach proves to be more efficient than previous models, and It allows for the development of indistinguishability under adaptive chosen ciphertext attack secure PRE scheme in the standard model, which was later achieved by Wang *et al.* [15] in 2015, utilizing Cramer-Shoup encryption. Their work was compared to that of Canetti and Hohenberger [16] in terms of efficiency. To address the certification management issue in PRE, Green and Ateniese [17] used classic PRE in identity-based cryptographic primitives for the first time. They also proposed two IB-PRE schemes, one of which is single-hop CCA secure and the other multi-hop CCA safe. In 2014, Yang *et al.* [18] presented the initial pairing-free CL-PRE technique. They demonstrated the absence of secrecy in Xu *et al.*'s CL-PRE scheme. They evaluated the computational efficiency of their schemes in comparison to those of Xu *et al.* [19] and Sur *et al.* [20].

The scheme proposed in this paper differs from the previous works in several ways. First, we use a one-way identifier-based proxy re-encryption protocol. We assume a cloud-resident semi-trusted proxy. Besides, we combine the re-encryption algorithm with a searchable encryption algorithm for efficient data screening. The adversary model and security analysis provided in this paper are also among this paper's

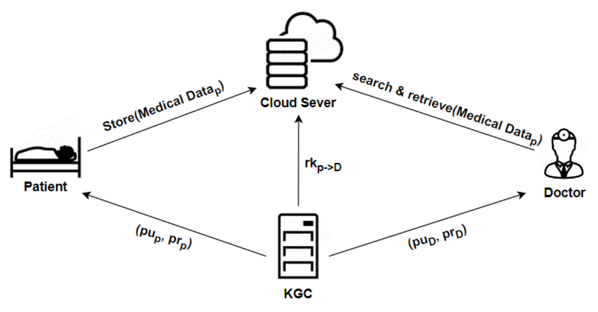

Figure 1. Architecture of system

contributions.

3 The System Model

3.1 System Architecture

Our system generally includes four entities: patient (data owner), key generation center, cloud service provider, and doctor (data receiver), as shown in Figure 1. The data owner is the patient who wants to store his medical data, which includes information related to the medical record or vital data obtained from sensors, in the cloud service server. In our system model, the patient is considered a trusted person. The cloud service provider is responsible for storing the medical data and is not authorized to access the medical data, and this entity is considered trusted but curious. The key generation center(KGC) is a trusted entity that generates public/private key pairs and sends the private key through a secure channel and the public key through a public channel to cloud users. Moreover, KGC generates a re-encryption key and transmits it to the proxy re-encryption server. The data recipients are doctors who want to access the medical data related to their patients that is stored on the cloud. This entity is untrusted in our system.

3.2 Adversary Model

We suppose two distinct sorts of attackers, each with different capabilities, aiming to compromise system confidentiality. The first kind of attacker poses as a malicious user accessing the exchanged messages in the system. We further assume he does not have access to the data stored in the cloud. This kind of attacker may be able to modify the messages exchanged over the network. As such, it may be able to change the public key sent to the entities by the KGC, but it cannot get or modify the principal secret keys. The second kind of attacker poses as an inquisitive cloud service provider with complete access to stored data. However, it cannot change any entity's public keys. Our suggested method ensures that none of these attackers can undermine the confidentiality of the messages.

4 Proposed Scheme

4.1 Preliminaries

An elliptic curve defined over finite field G is as $y^2 = x^3 + ax + b$. An intriguing property of elliptic curves is that they lead to a group structure, where a and b are elements from G that must satisfy the condition $4a^3 + 27b \neq 0$ to prevent multiple roots. The set's components are the points on the curve, and some action between the points is necessary to establish a group structure. Point addition and multiplying are the two potential geometrical operations specified over elliptic curve groups [21]. Based on the discrete logarithm problem (DLP) for elliptic curves, elliptic curves represent a significant area of cryptography. It claims it is difficult to discover the integer k for which $Q = kP$ is given two points, P and Q, on a curve. The action in this equation is a scalar value point multiplication $(P)(k)$ [22].

4.2 The Procedure

The ten algorithms listed below make up the overall syntax of the proposed scheme.

(1) **Initial setup:** The Key Generation Center is in charge of this algorithm. The algorithm outputs a principal secret key MK_{sec}, a principal public key MK_{pub}, and a set of public parameters Υ after receiving a secret parameter b as input.

(2) **Create fragmentary key:** A certain algorithm executed by Key Generation Center creates a fragmentary public and private key pair (P_{par_u}, S_{par_u}) that corresponds to the user's identity ID_u. This algorithm uses the public parameters Υ, cloud user biometric identifier BIO_u, principal secret key MK_{sec}, and principal public key MK_{pub} as inputs.

(3) **Create user key:** Every cloud user executes an algorithm that takes as inputs the user's biometric identification BIO_u and the public parameters Υ and produces the public key p_u and secret key k_u for users.

(4) **Create private key:** This procedure, which receives as inputs public parameters Υ, a fragmentary private key S_{par_u}, and the user's secret key k_u, and return as output the user's complete private key Sk_u, is executed by every cloud user u with a biometric identity BIO_u.

(5) **Create public key:** Every cloud user u with identification BIO_u performs this algorithm, which uses the user's secret key k_u, public parameters Υ, fragmentary public key P_{par_u}, and public key p_u as inputs to create a complete public key Pk_u for user u.

(6) **Encrypt:** A plaintext message m containing

a keyword w is encrypted using this algorithm by the data owner o using its public key Pk_o to create primary ciphertext c_o, which is then uploaded to the cloud.

(7) **Create re-encryption key:** This algorithm requires the data owner's public/private key pair (Pk_o, Sk_o) and the data recipient's public key (Pk_r) as inputs and generates the re-encryption key $rk_{o\rightarrow r}$ as output. The cloud resident proxy server receives a re-encryption key generated by the key generation center, $rk_{o\rightarrow r}$.

(8) **Re-encryption:** This algorithm is carried out by the proxy server, which converts primary ciphertext c_o obtained from the data owner o into secondary ciphertext c_r for the data receiver r using $rk_{o\rightarrow r}$ or returns symbol e if c_o is invalid.

(9) **Search:** This algorithm is run by a cloud server that takes the keyword w_o from the data owner and the keyword w_r from the data receiver and then searches among ciphertexts in its storage space [23]. Then, in case of matching the data receiver keyword with the stored ciphertexts keywords, returns ciphered message c_r to the data receiver or returns an error symbol e if the keyword does not match.

(10) **Decrypt:** This algorithm is carried out by the data receiver r, which receives the ciphertext c_r and decrypts it using its private key Sk_r to produce the appropriate plaintext message m or, in case of invalid c_r, an error symbol e.

4.3 LIPRE Scheme

The steps of the invented scheme are explained in this section.

- **Setup:** A security parameter b, a b-bit prime, and an elliptic curve E/F_p over a prime finite field F_p are all chosen by KGC. Let G represent the cyclic subgroup of an elliptic curve on E, with P serving as its generator. Additionally, KGC selects the principal secret key $s \in Z_q^*$ and the principal public key $K_{Pub} = sP$. It also selects the collision-free cryptographic hash algorithm H. The public parameters $\Upsilon = \{E, F_p, G, P, H, K_{Pub}\}$ are returned by this algorithm for publication.
- **Create fragmentary key:** KGC randomly selects x_{A_1}, x_{A_2}, and v_A and calculates $X_{A_1} = x_{A_1}P, X_{A_2} = x_{A_2}P, V_A = v_AP$. As a result, $S_{A_1} = (x_{A_1} + sH(H(BIO), X_{A_1})), S_{A_2} = (x_{A_2} + sH(H(BIO), X_{A_2}))$ and $A = v_A + sH(H(BIO_A), V_A, X_{A_1}, X_{A_2})$ are calculated. KGC sends to entity u, $X_A = (X_{A_1}, X_{A_2}, V_A, A)$ as a fragmentary public key over a public channel, and $S_A = (S_{A_1}, S_{A_2})$ as a fragmentary private key over a secure channel.
- **Set secret value:** Each identity u selects y_{A_1} and $y_{A_2} \in Z_q^*$ at random to serve as secret values for BIO_u.
- **Create private key:** This algorithm takes as inputs the user secret key (y_{A_1}, y_{A_2}), fragmentary private key S_A, and public parameters Υ and outputs a complete private key $Sk_A = (S_{A_1}, S_{A_2}, y_{A_1}, y_{A_2})$ for identity BIO_u.
- **Create public key:** This algorithm takes the user's secret key (y_{A_1}, y_{A_2}), fragmentary public key X_A, and public parameters Υ as inputs, computes $u_{A_1} = y_{A_1}P, u_{A_2} = y_{A_2}P$ and creates complete public key $Pk_A = (X_{A_1}, X_{A_2}, V_A, A, u_{A_1}, u_{A_2})$ for identity BIO_u.
- **Create re-encryption key:** The identity BIO_A, the identity BIO_B, the pair of public/private key (Pk_A, Sk_A) of the data owner, and the public key Pk_B of the receiver are all inputs for this algorithm. Re-encryption key $rk_{A\rightarrow B}$ is generated as follows:
  - ∘ $d_B = X_{B_1} + H(H(BIO_B), X_{B_1})K_{Pub}$
  - ∘ $d_{AB} = H(y_{A_1}d_B, S_{A_1}u_{B_2}, H(BIO_A), H(BIO_B), Pk_A, Pk_B)$
  - ∘ $rk_{A\rightarrow B} = ((S_{A_1} + y_{A_1})H(X_{A_1}, X_{A_2}, u_{A_1}, u_{A_2}) + S_{A_2} + y_{A_2})d_{AB}$
- **Encrypt:** It uses A's public key Pk_A, message m, keyword w, and public parameters Υ as inputs to generate the primary ciphertext c_A.
  (1) It confirms that $S_{A_1}P = X_{A_1} + H(H(BIO_u), X_{A_1})K_{Pub}$ and $S_{A_2}P = X_{A_2} + H(H(BIO_u), X_{A_2})K_{Pub}$ are valid fragmentary private keys and returns e if it is invalid.
  (2) It verifies whether any identity's public keys are valid. As is represented, $\beta_iP = V_i + H(H(BIO_i), V_i, X_{i_1}, X_{i_2})K_{Pub}$. Returns the error symbol e if it is incorrect.
  (3) It selects $\sigma \in \{0, 1\}^n$ and $\mu \in z_q^*$ and calculates $t = H(m, w, \sigma, BIO_A, u_{A_1}, u_{A_2})$.
  (4) It computes $c_1 = tP, c_3 = \mu P, c_2 = (m||\sigma) \oplus H(t((X_{A_1} + H(H(BIO_A), X_{A_1})K_{Pub} + u_{A_1})H(X_{A_1}, X_{A_2}, u_{A_1}, u_{A_2}) + X_{A_2} + H(H(BIO_A), X_{A_2})K_{Pub} + u_{A_2}))$, and $c_4 = \mu + tH(c_1, c_2, c_3)$. it returns $c_A = (c_1, c_2, c_3, c_4)$ For the proxy server.
- **Proxy re-encrypt:** The primary ciphertext c_A, the public parameter Υ, and the re-encryption key $rk_{A\rightarrow B}$ are all inputs to the cloud-resident proxy server. The received ciphertext is first verified as $c_4P = c_3 + H(c_1, c_2, c_3)c_1$. If successfully verificate, it generates $c_1' = c_1rk_{A\rightarrow B}$ and $c_2' = c_2$ and then returns the re-encrypted ciphertext $c_B = (c_1', c_2')$ for the receiver B. It returns the symbol e if not.
- **Search:** The cloud server runs this algorithm. Takes keywords w_A and w_B from the parties and searches among stored ciphertexts in the

cloud storage as follows:

(1) SrchEnc algorithm takes the secret key Sk_A and the keyword w_A from the data owner A as input and Encrypt them and outputs $I_A = SrchEnc(Sk_A, w_A)$.
(2) Transform algorithm takes the ciphertext I_A and the re-encryption key $rk_{A\rightarrow B}$ as input and returns $\tilde{I}_A = SrchTran(rk_{A\rightarrow B}, I_A)$ as output.
(3) This algorithm creates a trapdoor, taking the secret key Sk_B and the keyword w_B from data receiver B as input. $T_B = TrapCreat(Sk_B, w_B)$.
(4) Transform algorithm takes the trapdoor T_B and the re-encryption key $rk_{B\rightarrow A}$ as input and outputs $\tilde{T}_B = TrapTran(rk_{B\rightarrow A}, T_B)$.
(5) This part runs a matching searchable encryption algorithm based on Index, and the match function $(\tilde{I}_A, \tilde{T}_B)$ returns 1 in case of matching these two values and, otherwise, returns 0.

- **Decrypt:** It accepts as inputs the public parameters Υ, the cloud user u's private key Sk_u, and the ciphertext c_u, where $u \in \{A, B\}$ corresponds to user u. It then generates the corresponding plaintext message m; if c_u is incorrect, it returns an error symbol e.
  - $Decrypt_1$: The data owner A calculates $(m||\sigma) = c_2 \oplus H(((S_{A_1} + y_{A_1})H(X_{A_1}, X_{A_2}, u_{A_1}, u_{A_2}) + S_{A_2} + y_{A_2})c_1)$ to decrypt primary ciphertext $c_A = (c_1, c_2)$ using $A = (S_{A_1}, S_{A_2}, y_{A_1}, y_{A_2})$. If $c_1 = t'P$ holds, where $t' = H(m, w, \sigma, BIO_A, u_{A_1}, u_{A_2})$ returns plaintext m; otherwise, it yields an error symbol e.
  - $Decrypt_2$: To decrypt secondary ciphertext $c_B = (c'_1, c'_2)$ with $SK_B = (S_B, y_{B_1}, y_{B_2})$, data receiver B computes:
    - $d_A = X_{A_1} + H(H(BIO_A), X_{A_1})K_{Pub}$
    - $d_{BA} = H(u_{A_1}S_{B_1}, d_A y_{B_2}, H(BIO_A), H(BIO_B), Pk_A, Pk_B)$
    - $(m||\sigma) = c'_2 \oplus H((c'_1)/d_{BA})$

    and returns a plaintext message m if $(((X_{A_1} + H(H(BIO_A), X_{A_1})K_{Pub} + u_{A_1})H(X_{A_1}, X_{A_2}, u_{A_1}, u_{A_2}) + X_{A_2} + H(H(BIO_A), X_{A_2})K_{Pub} + u_{A_2})d_{AB}) = c'_1$ holds; otherwise, it returns an error symbol e.

5 Security Analysis

5.1 Informal Security Analysis

The proposed scheme meets forward security due to compliance with key separation and the use of separate keys in different stages of the proxy re-encryption process, and the attacker cannot decrypt previously re-encrypted data even with the private key of the delegator or proxy at a later time. Moreover, the proxy only possesses the necessary information to transform the data from the delegator's encryption to the delegatee's encryption and cannot recover any of the encryption keys used by the delegator or delegatee, thus ensuring that forward secrecy is maintained.

The keys used for re-encryption are kept separate from the current private keys, and the proxy cannot access the user's current private key, and the re-encryption keys are distinct from the user's current private key to maintain backward secrecy. In this way, backward security is ensured, and even if the attacker obtains the current private key of the protocol entities, the confidentiality of the previously stored data is maintained.

5.2 Formal Security Proof with Random Oracle Model

The formal proof represents that the suggested scheme is provably secure versus adversary $\mathcal{A}$, which wants to obtain the patient's identity (BIO_u), the secret and public key of the patient, the re-encryption key, and the plaintext message of the patient. The collision-resistant one-way hash function and ECDLP are two computationally hard issues. In this method, a mathematical proof is provided to demonstrate that the security of the proposed scheme is reduced to the adversary's ability to solve these two problems. First, we provide two definitions to prove the security of our proposed scheme in ROM.

Definition 1. The probability that the adversary $\mathcal{A}$ would arbitrarily choose the pair (x_1, x_2) within polynomial time t_1 such that $x_1 \neq x_2$ and $h(x_1) = h(x_2)$, as stated formally below, is an advantage for the adversary in finding a collision.

$$Adv_{\mathcal{A}}^{Hash}(t_1) = Pr[(x_1, x_2) \Leftarrow \mathcal{A}\colon x_1 \neq x_2 \wedge h(x_1) = h(x_2)] \quad (1)$$

If $Adv_{\mathcal{A}}^{Hash}(t_1) \leq \varepsilon_1$, the one-way hash function $h(.)$ is collision-resistant for every sufficiently small negligible function $\varepsilon_1 > 0$.

Definition 2. The elliptic curve discrete logarithm problem (ECDLP) asks for the determination of the integer $s \in Z_p^*$ from two points $P, Q(= [s]P) \in E(F_p)$. While solving the ECDLP during execution time t_2, adversary $\mathcal{A}$ has an advantage described as

$$Adv_{\mathcal{A}}^{ECDLP}(t_2) = Pr[s \in \mathbb{Z}_p^* : P, Q = [s]P \in E(\mathbb{F}_p)] \quad (2)$$

Every sufficiently tiny negligible function $\varepsilon_2 > 0$ and any probabilistic polynomial time-bounded algorithm $\mathcal{A}$ are intractable if $Adv_{\mathcal{A}}^{ECDLP}(t_2) \leq \varepsilon_2$.

This formal proof assumes that adversary $\mathcal{A}$ has abilities noted in Section 3.2. In addition, adversary $\mathcal{A}$ has access to the following oracles:

- $Reveal(H(M))$: Consider the one-way hash function $H(M)$; Oracle will categorically yield the value M.
- $Extract(Q, P)$: Consider the input P and $Q = [s]P$; the oracle categorically yields the secret value s.

Theorem 1. *Considering that ECDLP is a computationally intractable issue and that the cryptographic one-way hash function $h(.)$ behaves like a real random oracle. When obtaining the patient's identifier BIO_i, the secret parameter s, re-encryption key $rk_{A\to B}$, and message m, the proposed scheme is provably secure against adversary $\mathcal{A}$.*

Proof. Assume the adversary $\mathcal{A}$ is built to perform the algorithm $ALG^{Hash,ECDLP}_{LIPRE,\mathcal{A}}$, as given in Algorithm 1, for the proposed proxy re-encryption protocol to determine the patient's identifier BIO_i, secret key s, re-encryption key $rk_{A\to B}$, and content of message. Based on the assumption that the adversary $\mathcal{A}$ will be able to get the sent messages and parameters over public channel. Hence, $Succ^{Hash,ECDLP}_{LIPRE,\mathcal{A}} = 2Pr[Adv^{Hash,ECDLP}_{LIPRE,\mathcal{A}} = 1] - 1$ expresses the likelihood that $ALG^{Hash,ECDLP}_{LIPRE,\mathcal{A}}$ will succeed. The advantage for the $ALG^{Hash,ECDLP}_{LIPRE,\mathcal{A}}$ is the maximum success probability taken across all $\mathcal{A}$ with execution time t, $Adv^{Hash,ECDLP}_{LIPRE,\mathcal{A}}(t, q_1, q_2) = max_{\mathcal{A}}\{Succ^{Hash,ECDLP}_{LIPRE,\mathcal{A}}\}$, where q_1 and q_2 represent the number of queries performed to the Oracles *Reveal* and *Extract*, respectively.

Assume, based on algorithm $ALG^{Hash,ECDLP}_{LIPRE,\mathcal{A}}$, that adversary $\mathcal{A}$ may use the oracles Extract and Reveal to solve ECDLP and compute the inverse of cryptographic one-way hash functions. After that, adversary $\mathcal{A}$ wins in the game and successfully acquires BIO_i, secret parameter s,re-encryption key $rk_{A\to B}$, and message m. The advantages $Adv^{Hash}_{\mathcal{A}}(t_1) \le \varepsilon_1$ and $Adv^{ECDLP}_{\mathcal{A}}(t_2) \le \varepsilon_2$ for all sufficiently small negligible functions $\varepsilon_1, \varepsilon_2 > 0$, however, are stated in Definitions 1, 2. Moreover, because every sufficiently tiny $\varepsilon > 0$, it follows that $Adv^{Hash,ECDLP}_{LIPRE,\mathcal{A}}(t, q_1, q_2) \le \varepsilon$. As a result, Theorem 1 is proved. □

In this way, it was shown that the plan's security was first reduced to the security of proxy re-encryption. Then, the security of proxy re-encryption was reduced to the difficulty of the discrete logarithm in the elliptic curve.

Algorithm 1 . $ALG^{Hash,ECDLP}_{LIPRE,\mathcal{A}}$

```
1: After the KGC sends parameters $\Upsilon$ to the protocol entities, the adversary calls Oracle extract and calculates the secret value $s.\longleftarrow extract(P_{pub}, P)$
2: Due to sending $R_A$ on the public channel, the adversary can obtain private values, including $BIO_u, r_{A_1}, r_{A_2}$ and $v_A$, by calling extract and reveal oracles.$\longleftarrow extract(R_{A_1}, P), reveal(H(H(BIO_u), V_A, R_{A_1}, R_{A_2}))$ and so on.
3: Having the $Pk_A$ parameter and the values obtained from the previous steps, The adversary can calculate the patient's private key. $\longleftarrow compute(Sk_A)$.
4: Similarly, the adversary calculates the values of $d_B, d_{AB}$, and re-encryption key $rk_{A\to B}$ for the receiver by obtaining the above values. $\longleftarrow compute(rk_{A\to B})$.
5: if $S_{A_1}P = S'_{A_1}P$ then
6:     The adversary gets the first validation for encrypting the message.
7:     if $\beta_i P = \beta'_i P$ then
8:         The adversary gets the second validation for encrypting the messages.
9:         if $c_1 = t'P$ then
10:            The adversary for first-level decryption is valid.
11:            if $c_1 = c'_1$ then
12:                The adversary for secondary decryption is valid and can decrypt ciphertext messages.
13:                return (Success)
14:            else
15:                return (Fail)
16:            end if
17:        else
18:            return (Fail)
19:        end if
20:    else
21:        return (Fail)
22:    end if
23: else
24:    return (Fail)
25: end if
```

6 Performance Comparison

In this part, we assess and compare our suggested scheme with three other schemes. To evaluate the schemes in terms of execution time, we considered a specific execution time for each operation used in the proxy re-encryption schemes. Table 1 shows the considered execution time for each operation. Execution time of exponentiation, bilinear pairing, point multiplication, point addition, and modular inversion operations are 5.31, 16.39, 2.184, 0.22, and 5.16 milliseconds, respectively. We are ignoring the calculation cost associated with these operations because general hash operations and point additions take very little time to compute compared to other operations. We compared our proposed scheme with Yang *et al.* [18], Osama [24] and Eman [25] proxy re-encryption schemes, and with the help of the execution times in Table 1 and the number of performed operations in each phase of the protocol, we obtained the total execution time of each scheme. Table 2 shows the number of operations and computed execution time for each scheme in that LIPRE achieved the best

Table 1. Notations used in computation cost analysis

symbol	description	execution time(ms)
t_e	time of exponentiation operation	5.31
t_{bp}	time of bilinear pairing operation	16.39
t_{pm}	time of ECC point multiplications	2.184
t_{am}	time of ECC point addition	0.22
t_{inv}	time of modular inversion	5.16

Table 2. Performance comparison

scheme / algorithm	Yang [18]	Osama [24]	Eman [25]	proposed scheme
Initial setup	t_e	t_{bp}	t_{pm}	t_{pm}
key generation	$6t_e$	$3t_{pm} + t_{inv}$	$3t_{pm} + t_{inv}$	$9t_{pm}$
Encrypt	$4t_e$	$2t_e + t_{inv}$	$2t_{pm} + t_{am}$	$5t_{pm}$
Re-encryption	t_e	$3t_{bp}$	t_{pm}	t_{pm}
Decrypt1	$3t_e$	$t_e + t_{inv}$	$t_e + t_{inv}$	$2t_{pm}$
Decrypt2	t_e	$4t_e + t_{am}$	$4t_e + t_{am}$	$5t_{pm}$
Total computation time(ms)	84.96	118.43	52.598	50.232
Assumption	CDH	p-BDHI	EC-CDH	EC-CDH
Attacked	✓	×	×	×

Note: Abbreviations: CDH, computational Diffie-Hellman; p-BDHI, p-bilinear Diffie-Hellman inversion ; EC-CDH, elliptic curve computational Diffie-Hellman .

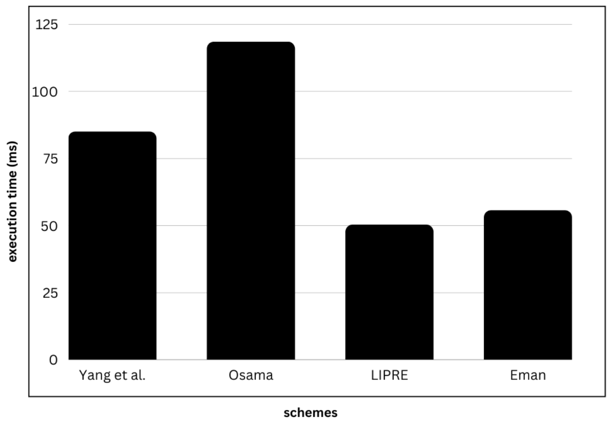

Figure 2. Diagram of time execution comparison

runtime compared to other schemes. Moreover, there are attacks on Yang *et al.* [18] scheme, but not discovered attacks on two other schemes, and our scheme resists attacks, too. The assumption of each scheme is shown in the table also. Figure 2 shows the comparison diagram of the execution time of the schemes.

7 Conclusion

Secure medical personal health records storage and sharing while utilizing cloud resources for storing encrypted data without disclosing the contents of that message to the cloud proxy server is a crucial concern. The proxy re-encryption primitive promises to overcome these issues and provide secure data sharing in the cloud. A single-hop, pairing-free, unidirectional Identity-based PRE (LIPRE) scheme based on ECC has been proposed to exchange medical data in public clouds safely. In the random oracle paradigm, the suggested PRE method is proven safe. The suggested scheme is more computationally effective than current schemes and may be employed with existing mobile devices with limited resource availability in the Internet of Medical Things. In future works, we aim to implement our protocol in a real environment to evaluate its performance.

References

[1] S Vishnu, SR Jino Ramson, and R Jegan. Internet of medical things (iomt)-an overview. In *2020 5th international conference on devices, circuits and systems (ICDCS)*, pages 101–104. IEEE, 2020.

[2] Ali Ghubaish, Tara Salman, Maede Zolanvari, Devrim Unal, Abdulla Al-Ali, and Raj Jain. Recent advances in the internet-of-medical-things (iomt) systems security. *IEEE Internet of Things Journal*, 8(11):8707–8718, 2020.

[3] George Hatzivasilis, Othonas Soultatos, Sotiris Ioannidis, Christos Verikoukis, Giorgos Demetriou, and Christos Tsatsoulis. Review of security and privacy for the internet of medical things (iomt). In *2019 15th international conference on distributed computing in sensor systems (DCOSS)*, pages 457–464. IEEE, 2019.

[4] Mohammad Yaghoubi, Khandakar Ahmed, and Yuan Miao. Wireless body area network (wban): A survey on architecture, technologies, energy consumption, and security challenges. *Journal of Sensor and Actuator Networks*, 11(4):67, 2022.

[5] David Nuñez, Isaac Agudo, and Javier Lopez. Proxy re-encryption: Analysis of constructions and its application to secure access delegation. *Journal of Network and Computer Applications*, 87:193–209, 2017.

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

[7] Liang Xue, Yong Yu, Yannan Li, Man Ho Au, Xiaojiang Du, and Bo Yang. Efficient attribute-based encryption with attribute revocation for assured data deletion. *Information Sciences*, 479:640–650, 2019.

[8] Xuanmei Qin, Yongfeng Huang, Zhen Yang, and Xing Li. Lbac: A lightweight blockchain-based access control scheme for the internet of things. *Information Sciences*, 554:222–235, 2021.

[9] Mang Su, Bo Zhou, Anmin Fu, Yan Yu, and Gongxuan Zhang. Prta: A proxy re-encryption based trusted authorization scheme for nodes on cloudiot. *Information Sciences*, 527:533–547, 2020.

[10] Xiaohui Liang, Zhenfu Cao, Huang Lin, and

Jun Shao. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 4th international symposium on information, computer, and communications security*, pages 276–286, 2009.

[11] Chunpeng Ge, Zhe Liu, Jinyue Xia, and Liming 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.

[12] Ryuichi Sakai and Jun Furukawa. Identity-based broadcast encryption. *Cryptology ePrint Archive*, 2007.

[13] Cheng-Kang CHU, Jian Weng, Sherman SW Chow, Jianying Zhou, and Robert H DENG. Conditional proxy broadcast re-encryption.(2009). In *Information Security and Privacy: 14th Australasian Conference, ACISP*, pages 1–3, 2009.

[14] Robert H Deng, Jian Weng, Shengli Liu, and Kefei Chen. Chosen-ciphertext secure proxy re-encryption without pairings. In *Cryptology and Network Security: 7th International Conference, CANS 2008, Hong-Kong, China, December 2-4, 2008. Proceedings 7*, pages 1–17. Springer, 2008.

[15] Xu An Wang, Jianfeng Ma, and Xiaoyuan Yang. A new proxy re-encryption scheme for protecting critical information systems. *Journal of Ambient Intelligence and Humanized Computing*, 6:699–711, 2015.

[16] Ran Canetti and Susan Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 185–194, 2007.

[17] Matthew Green and Giuseppe 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.

[18] Kang Yang, Jing Xu, and Zhenfeng Zhang. Certificateless proxy re-encryption without pairings. In *Information Security and Cryptology–ICISC 2013: 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers 16*, pages 67–88. Springer, 2014.

[19] Lei Xu, Xiaoxin Wu, and Xinwen Zhang. Cl-pre: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In *Proceedings of the 7th ACM symposium on information, computer and communications security*, pages 87–88, 2012.

[20] Chul Sur, Chae Duk Jung, Youngho Park, and Kyung Hyune Rhee. Chosen-ciphertext secure certificateless proxy re-encryption. In *Communications and Multimedia Security: 11th IFIP TC 6/TC 11 International Conference, CMS 2010, Linz, Austria, May 31–June 2, 2010. Proceedings 11*, pages 214–232. Springer, 2010.

[21] Lawrence C Washington. *Elliptic curves: number theory and cryptography*. CRC press, 2008.

[22] Darrel Hankerson, S Vanstone, and A Menezes. Guide to elliptic curve cryptography. *Springer Science And Business Media*, 2006.

[23] Nitish Andola, Raghav Gahlot, Vijay Kumar Yadav, S Venkatesan, and Shekhar Verma. Searchable encryption on the cloud: a survey. *The Journal of Supercomputing*, 78(7):9952–9984, 2022.

[24] Osama A Khashan. Hybrid lightweight proxy re-encryption scheme for secure fog-to-things environment. *IEEE Access*, 8:66878–66887, 2020.

[25] Eman Abouelkeir. Provable lightweight hybrid proxy re-encryption scheme without pairings for internet of things. 2022.

Iman Jafarian received a bachelor's degree in Computer Engineering from Yazd University, Iran, in 2021. He is a master science student in Secure Computing at Amirkabir University of Technology, Tehran, Iran, since 2021. His research areas include Information Security, Cryptography, and Security Protocols.

Siavash Khorsandi received the B.Sc. and M.Sc. degrees in Electrical Engineering from Tehran Polytechnic, in 1987 and 1989, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Toronto, in 1996. He joined Amirkabir University, in 2001, after a four-year Tenure, from 1996 to 2000, at Nortel Networks working as a Senior Engineer on Advanced Network Architectures. He is currently an Associate Professor with the Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran.