mathematics

MDPI

Article

Design of Secure and Privacy-Preserving Data Sharing Scheme Based on Key Aggregation and Private Set Intersection in Medical Information System

Jihyeon Oh [1], Seunghwan Son [1], DeokKyu Kwon [1], Myeonghyun Kim [1], Yohan Park [2] and Youngho Park [1,*]

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; j2hnoh@knu.ac.kr (J.O.); sonshawn@knu.ac.kr (S.S.); kdk145@knu.ac.kr (D.K.); kimmyeong123@knu.ac.kr (M.K.)

[2] School of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; yhpark@kmu.ac.kr

* Correspondence: parkyh@knu.ac.kr

Abstract: Medical data sharing is pivotal in enhancing accessibility and collaboration among healthcare providers, researchers, and institutions, ultimately leading to enhanced patient outcomes and more efficient healthcare delivery. However, due to the sensitive nature of medical information, ensuring both privacy and confidentiality is paramount. Access control-based data sharing methods have been explored to address these issues, but data privacy concerns still remain. Therefore, this paper proposes a secure and privacy-preserving data sharing scheme that achieves an equilibrium between data confidentiality and privacy. By leveraging key aggregate encryption and private set intersection techniques, our scheme ensures secure data sharing while protecting against the exposure of sensitive information related to data. We conduct informal and formal security analyses, including Burrow–Abadi–Needham logic and Scyther, to demonstrate its resilience against potential adversarial attacks. We also implement the execution time for cryptographic operations using multiprecision integer and a rational arithmetic cryptographic library and perform comparative analysis with existing related schemes in terms of security, computational cost, and time complexity. Our findings demonstrate a high level of security and efficiency, demonstrating that the proposed scheme contributes to the field by providing a solution that protects data privacy while enabling secure and flexible sharing of medical data.

Keywords: medical data sharing; key aggregate encryption; private set intersection; homomorphic encryption; mutual authentication

MSC: 68M12

Citation: Oh, J.; Son, S.; Kwon, D.; Kim, M.; Park, Y.; Park, Y. Design of Secure and Privacy-Preserving Data Sharing Scheme Based on Key Aggregation and Private Set Intersection in a Medical Information System. *Mathematics* **2024**, *12*, 1717. https://doi.org/10.3390/math12111717

Academic Editors: Cheng-Chi Lee and Dinh-Thuan Do

Received: 3 May 2024
Revised: 29 May 2024
Accepted: 30 May 2024
Published: 31 May 2024

1. Introduction

With the rapid advancement of modern technologies and the increasing digitalization of the medical sector, there has been a significant surge in both the volume and diversity of medical data. This proliferation, especially within the realm of medical information systems such as electronic health records (EHRs) and health information exchange (HIE), promotes accessibility and data sharing among healthcare providers, researchers, and institutions, improving patient outcomes, accelerating healthcare discovery, and optimizing healthcare delivery [1,2]. Medical data can be used to identify patterns and correlations to advance the understanding, diagnosis, and treatment of disease. This can improve quality of care and patient satisfaction by personalizing the care process and effectively managing chronic conditions. Ultimately, medical data sharing is an essential element of modern healthcare and plays an important role in advancing medical information systems and improving people's quality of life.

Despite the numerous advantages of medical data sharing, significant concerns remain, particularly in the realms of security and privacy. The sensitive nature of medical information means that without adequate protection, there is a substantial risk of severe privacy violations and data breaches [3–5]. Unauthorized access or the theft of medical data can not only infringe upon privacy rights but also undermine trust in the entire medical information system. Data breaches can disrupt healthcare operations, compromise the integrity of research efforts, and even jeopardize public health initiatives. Ensuring the security of medical data is not merely about privacy but is also a critical aspect of protecting the entire healthcare infrastructure. This underscores the crucial importance of balancing confidentiality and privacy in data sharing, prompting the exploration of advanced solutions that can enhance utility while ensuring data security. Hence, it is imperative to conduct research on secure sharing of medical data by restricting access to information.

To enhance medical data security, researchers have increasingly explored data sharing frameworks that leverage access control technologies, such as attribute-based encryption (ABE) [6] and key aggregate encryption (KAE) [7]. Based on specific properties for generating and decrypting ciphertext, these frameworks ensure that only authorized users can access data, allowing data owners to securely share sensitive data while maintaining strict control over access rights. While offering substantial flexibility in data and access rights management, they also raise serious concerns regarding data privacy, necessitating thorough examination from alternative perspectives. In access control-based systems, privacy vulnerabilities can arise from either by revealing sensitive information through attribute values [8–10] or by exposing the data-related information itself that was caused by data users having to identify the desired data before requesting an access key from the owner. Conversely, if data owners choose to withhold data-related information, data users will resort to inefficient and insecure practices, such as randomly querying on a cloud server to determine the availability of certain data. This method is inherently flawed as it neither guarantees efficiency nor meets the security requisite for handling sensitive medical data. Therefore, there is a need for further research and development in medical data sharing methodologies that enable secure data sharing between data owners and users while maintaining data privacy.

In this paper, we propose a design of a secure and privacy-preserving data sharing scheme for medical information systems. We integrate private set intersection (PSI) with KAE to achieve an equilibrium between data confidentiality and privacy. To ensure secure and adaptable access control over medical data, we leverage a single access key feature of KAE [11] while integrating PSI to alleviate potential privacy concerns. PSI enables both data owners and users to confirm the presence of common information in their respective private sets without revealing the information about them, allowing them to issue or request access keys only after verifying intersection. This mechanism markedly diminishes the necessity for data owners to divulge data-related information, effectively mitigating a primary privacy concern. By adopting this approach, we not only reduce the risk of information exposure but also fortify the overall data security framework within medical information systems and address the imperatives of data confidentiality and privacy preservation in medical data sharing. The key considerations of this paper are as follows.

- We propose a privacy-preserving medical data sharing scheme. To maintain a balance between privacy and data sharing, we leverage PSI between the data owner and the data user before access requests. This facilitates interaction and data sharing while protecting sensitive information.
- The proposed scheme ensures secure data sharing and access control through KAE. Since KAE enables secure and flexible access control with a single aggregate key, the integration of KAE in the proposed scheme enhances data security by reducing the risk of data breaches and unauthorized disclosures.
- We perform security analysis using the Scyther tool [12] and mathematical analysis methods such as Burrows–Abadi–Needham (BAN) logic [13] and indistinguishability against the chosen plaintext attack (IND-CPA). In addition, we conduct performance

analyses using the multiprecision integer and rational arithmetic cryptographic library (MIRACL) [14] and compare the obtained results with those of previous studies.

The remainder of this paper is organized as follows. The related works are presented in Section 2, and the preliminaries for the paper are in Section 3. Section 4 presents the system model for the proposed scheme, including network model, adversary model, and security model. Section 5 explains the proposed medical data sharing scheme. Informal and formal Security analyses are performed in Section 6, and the comparative analysis is conducted in Section 7. Section 8 summarizes the conclusions of this paper.

2. Related Works

Considerable research has been conducted in the realm of medical data sharing with a notable focus on data security and privacy preservation. In 2022, Bao et al. [15] proposed a lightweight ABE scheme, specifically tailored for the Internet of Things (IoT) and supported by cloud technology, within smart healthcare systems. Their approach prioritizes both the efficiency required by resource-limited devices and the implementation of fine-grained access control, ensuring that data access aligns with user authorization levels. Mamta et al. [16] developed a secure and efficient fine-grained data sharing scheme for IoT-based healthcare systems. Their approach critiques existing models of fine-grained medical data sharing and leverages fog computing alongside ABE to significantly reduce the computational load on data users while simultaneously enhancing data confidentiality. Wang et al. [17] proposed a consortium blockchain-based scheme for personal health record (PHR) management and sharing that prioritizes both security and privacy. They emphasized the importance of allowing patients to customize access control to their PHR according to their individual preferences, ensuring that only authorized users have access. To achieve this, they integrated a modified ABE scheme with smart contracts, enabling functionalities for secure search, privacy preservation, and personalized access control. Oh et al. [18] introduced a patient-centric secure PHR sharing system, addressing data integrity, transparency, mutual authentication, etc. They acknowledged the common use of ABSE in medical data sharing but identified key management challenges inherent in its implementation. To address this issue, they adopted the concept of key aggregate searchable encryption (KASE), presenting a key aggregate dynamic searchable encryption framework integrated with a linear secret sharing scheme.

In 2023, Trivedi and Patel [19] developed a KASE-based framework for sharing electronic health records in integrated healthcare systems on clouds. They identified limitations in existing schemes, particularly the lack of secure multi-user authorization and keyword untraceability. Their framework addresses these limitations by incorporating robust security features, including secure multi-user authorization and keyword untraceability. This approach not only enhances security but also demonstrates significant efficiency gains in storage, communication, and computation. Xu et al. [20] devised a privacy-enhanced medical data sharing framework that utilizes an authorization mechanism and ABE on the blockchain, aiming to address the challenges of fragmented healthcare systems, which can compromise treatment quality and lead to privacy breaches. Their approach empowers data owners with control over data access via ABE, complemented by an efficient authorized and revocable mechanism, ensuring access for authorization and revocation mechanism, which ensures that authorized doctors can access data while swiftly revoking access for unauthorized individuals. Zhang et al. [21] developed a multi-server search scheme that facilitates collaborative operations among various healthcare entities for tasks such as diagnostic institution location, medical data retrieval, and cross-domain data exploration. Their scheme incorporates a secure data transfer method that enables servers from disparate organizations to perform joint computational tasks while preserving the confidentiality of each participant's data and the privacy of their search identities. Zhang et al. [22] identified that while weighted ABE enhances the flexibility of access policies in medical data sharing, it poses challenges for data owners striving to maintain control over their privacy, particularly in collaborative e-health systems. Aiming to strike an optimal balance

between privacy and flexibility, they proposed a cloud-based system for sharing personal health records. This system employs AND-weighted ABE, enabling users to access data only if they belong to specified organizations, thus reinforcing both security and selective accessibility. Peng et al. [23] proposed a patient-centric EMR sharing scheme, aiming to tackle the complex issues of privacy concerns and inter-agency distrust stemming from the misuse of access to medical records and the challenge of admitting unconscious patients. They designed an architecture for privacy-preserving medical data sharing, leveraging a dual-blockchain system and an identity-based tripartite authentication key agreement scheme, fostering trust between patients and healthcare institutions.

In 2024, Zhang et al. [24] pointed out that existing attribute-based searchable encryption schemes could expose sensitive information about data users and lead to data tampering and even untrusted results due to the delegation of complex search operations to a cloud server. In response, they proposed a blockchain-based anonymous ABSE scheme to enhance data sharing security. This approach conceals the attributes of the access policy, thereby safeguarding the confidentiality of the attributes that fulfil the access requirements. By integrating ABSE with blockchain technology, the scheme also incorporates features like tamper-proofing, integrity verification, and non-repudiation, significantly bolstering the trust and security of digital transactions. Jastaniah et al. [25] introduced the SAMA scheme, crafted to overcome the shortcomings of current methodologies in managing data aggregation and sharing for wearable devices. Their objective was to offer a scheme that is not only centered around the user and privacy-friendly but also flexible enough to efficiently support multiple data owners and requesters. The SAMA scheme integrates multi-key partial homomorphic encryption with ciphertext-policy ABE to ensure robust data confidentiality, user-centric access control, and streamlined data processing tailored for wearable technology. Yin et al. [26] indicated the paucity of research into the privacy implications associated with user identity during the key generation phase. To address this gap, they proposed a decentralized ciphertext-policy ABE scheme, specifically designed to bolster the secure dissemination of sensitive healthcare information within blockchain-enabled healthcare systems. Leveraging Shamir's threshold secret sharing, their scheme distributed the master key across all attribute nodes in the blockchain, thereby augmenting the robustness and enhancing the system's resilience to adversarial attacks.

While existing research in medical data sharing has made significant progress in areas such as security, efficiency, and privacy, there remains a crucial aspect requiring more focused attention. Given the highly sensitive nature of medical data, the potential for information leakage through attributes and data-related information in access control-based systems, as observed in the aforementioned study, poses significant privacy concerns. This issue impedes the development of secure data sharing practices, subsequently restricting thorough data analysis and mutual efforts [27,28]. To enhance cooperation among various data owners and advance medical research, it is essential to prioritize the protection of each entity's data privacy. This entails limiting the information disclosed to only what is necessary during the process of sharing medical data. Hence, we utilize key aggregation encryption and private set intersection to protect data confidentiality and prevent any inadvertent exposure of sensitive information, and verify the legitimacy of entities through mutual authentication. This process protects data at all stages, including data upload, storage, transmission, and access and ensures a secure data sharing environment.

3. Preliminaries

This section briefly introduces the foundational mathematical and technical principles to aid in comprehending the contents of this document.

3.1. Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is a public key cryptography that exploits the mathematical properties of elliptic curves over finite fields [29]. An elliptic curve $E_p(a, b)$ over a finite field $\mathbb{Z}_p$ is defined as $E_p(a, b) : y^2 \equiv x^3 + ax + b \pmod{p}$, where p is a large

prime integers and $x, y, a, b \in \mathbb{Z}_p$, ensuring that the discriminant $4a^3 + 27b^2 \pmod{p} \neq 0$. The additive group $\mathcal{G} = \{(x, y) : x, y \in \mathbb{Z}_p, (x, y) \in E/\mathbb{Z}_p\} \cup \{\mathcal{O}\}$ is defined, where $\mathcal{O}$ symbolizes the point at infinity, serving as the identity element of $\mathcal{G}$. Scalar multiplication is defined by repeated addition operation as $\alpha P = P + P + \cdots + P$ (α times), with a base point $P \in \mathcal{G}$ and an integer $\alpha \in \mathbb{Z}_p^*$. The mathematical security of ECC are represented as follows.

- Elliptic curve discrete logarithm problem (ECDLP): Given two points P and Q on $E_p(a, b)$, determining the scalar $\alpha \in \mathbb{Z}_p$ such that $Q = \alpha \cdot P$ is considered computationally difficult.
- Elliptic curve computational Diffie–Hellman problem (ECCDHP): Given two points $\alpha \cdot P$ and $\beta \cdot P$, it is hard to calculate $\alpha \cdot \beta \cdot P$.
- Elliptic curve decisional Diffie–Hellman problem (ECDDHP): Given three points $\alpha \cdot P$, $\beta \cdot P$, and $\gamma \cdot P$, it is difficult to determine whether $\gamma \cdot P = \alpha \cdot \beta \cdot P$, where $\alpha, \beta, \gamma \in \mathbb{Z}_p$.

3.2. Bilinear Pairing

Let $\mathcal{G}$ be an additive group, consisting of points on an elliptic curve E defined over a field $\mathbb{F}$, having order n and identity element $\mathcal{O}$. Let $\mathcal{G}_T$ be a multiplicative group. A bilinear pairing $\hat{e} : \mathcal{G} \times \mathcal{G} \to \mathcal{G}_T$ satisfies the following conditions.

- Bilinearity: For $\forall P, Q \in \mathcal{G}$, and $\forall a, b \in Z_p^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
- Non-degeneracy: $\hat{e}(P, Q) \neq 1$ for some $P, Q \in \mathcal{G}$.
- Efficiency: $\hat{e}(P, Q)$ can be calculated in polynomial time for $\forall P, Q \in \mathcal{G}$.

3.3. Decisional Bilinear Diffie–Hellman (DBDH) Assumption

This assumption assumes that a probabilistic polynomial-time adversary $\mathcal{A}$ lacks the ability to differentiate between $(aP, bP, cP, \hat{e}(P, P)^{abc})$ and $(aP, bP, cP, \hat{e}(P, P)^{d})$. Consequently, we can express the $\mathcal{A}$'s advantage ε as follows.

$$|Pr[\mathcal{A}(aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] - Pr[\mathcal{A}(aP, bP, cP, \hat{e}(P, P)^{d}) = 1]| \geq \varepsilon$$

DBDH assumption holds if $\mathcal{A}$ cannot distinguish $\hat{e}(P, P)^{d} = \hat{e}(P, P)^{abc}$, i.e., whether $d = abc$ or $d \in \mathbb{Z}_q^*$, with an inescapable advantage.

3.4. Key Aggregate Encryption

Key aggregate encryption (KAE) is an access control cryptosystem that streamlines data decryption procedures by allowing the decryption of a collection of data encrypted with multiple keys using a single constant aggregate key [7]. This aggregate key, though as concise as a solitary secret key, combines the capabilities of numerous such keys, granting decryption authority for any subset of ciphertext classes. In contrast to traditional systems that require a distinct key for each ciphertext, KAE uses a single aggregate key, reducing complexity and cost. This approach not only diminishes the key management overhead but also amplifies efficiency in data sharing. However, most KAE schemes rely on bilinear operation, incurring significant computational overhead [30]. Especially in scenarios involving data sharing, processing, and transmission of large datasets, such methods prove to be inefficient. In response, an alternative approach called ECC-based KAE was introduced. This method optimizes resource utilization by leveraging the small key size of ECC, ensuring robust security while facilitating efficient data transmission and processing. Consequently, the proposed system adopts the ECC-based KASE method, with the operational process outlined as follows.

(1) KAE.Setup $(1^\lambda, n)$: Generate a random number $\alpha \in \mathbb{Z}_p$, compute $P_i = \alpha^i P \in \mathcal{G}$ for $i = \{1, \ldots, n, n+2, \ldots, 2n\}$, and publish $param = \{P, n, \{P_i\}_{1 \leq i \leq 2n, i \neq n+1}\}$. Then, discard α.

(2) KAE.KeyGen (): Generate $sk \in \mathbb{Z}_p$ and compute $pk = sk \cdot P$. Then, output public and private key pair $(pk, sk) = (sk \cdot P, sk)$.

(3) KAE.Encrypt ($param, pk, i, F$): For data $F_i \in \mathcal{G}_T$ in $i \in \{1, \ldots, n\}$, choose a random number $s \in \mathbb{Z}_p$ and compute $c_1 = s \cdot P$, $c_2 = s \cdot (pk + P_i)$, $c_3 = F_i \cdot e(P_1, P_n)^s$. Then, output $C = \{c_1, c_2, c_3\}$.

(4) KAE.Extrac t ($param, sk, S$): For the subset of data class indices S, output the aggregate key $AK = \sum_{j \in S} sk \cdot P_{n+1-j}$.

(5) KAE.Decrypt ($param, AK, S, i, C$): If $i \neq S$, output $\perp$. Otherwise, calculate $v_1 = \sum_{j \in S, j \neq i} P_{n+1-j+i}$, $v_2 = \sum_{j \in S} P_{n+1-j}$, and output $F_i = c_3 \cdot \frac{e(AK+v_1, c_1)}{e(v_2, c_2)}$.

3.5. Brakerski–Gentry–Vaikuntanathan

Brakerski–Gentry–Vaikuntanathan (BGV) [31] is a type of fully homomorphic encryption (FHE) that enables arithmetic operations on encrypted data. BGV eliminates the need for decryption, producing an encrypted output that, when decrypted, yields the same result as operations performed on the plaintext. The BGV scheme is renowned for its effectiveness in performing unlimited additions and multiplications on encrypted data, which is facilitated by a process known as bootstrapping. This process effectively manages the noise generated during computations, ensuring the encryption integrity. Below is a description of the BGV algorithm.

(1) BGV.Setup (1^λ): Select a ring $R_q = \mathbb{Z}_q[X]/(X^l + 1)$, where l is a power of 2. Given a security parameter λ, set the ciphertext modulus q, plaintext modulus t, and the noise distribution $\mathcal{X}$. Output $params = (R_q, l, q, t, \mathcal{X})$.

(2) BGV.KeyGen ($param$): Generate the secret key $s \in \{-1, 0, 1\}^l$, a random polynomial $r \in R_q$, and a random error polynomial $e \in \mathcal{X}$. Calculate the public key $p = (p_0, p_1) = (r \cdot s + t \cdot e, -r)$.

(3) BGV.Enc ($params, pk, m$): Generate $e_0, e_1 \in \mathcal{X}$, random polynomial u, and compute $\mathbf{ct} = (ct_0, ct_1) = (p_0 \cdot u + te_0 + m, p_1 \cdot u + te_1) = [\![m]\!]_p$.

(4) BGV.Dec ($params, s, \mathbf{ct}$): Calculate $m = \left[[ct_0 + ct_1 \cdot s]_q\right]_t$ using s.

3.6. Private Set Intersection

Private set intersection (PSI) is a cryptographic protocol designed to identify common elements between sets held by two or more parties, such as individuals or organizations, without revealing any underlying data. This functionality enables the parties to determine overlapping information while preserving the confidentiality of their respective datasets. In the typical PSI scenario discussed in this paper, the sender and receiver have sets X and Y with sizes N_X and N_Y, respectively. Upon the receiver's request for the intersection, the sender computes it and transmits the result. The receiver leverages their private key to decrypt the intersected set. The detailed structure of the PSI employed in this study is as follows.

(1) PSI.Setup (1^λ): Sender and receiver each generate a public–private key pair using the BGV.KeyGen procedure.

(2) PSI.Enc (Y, p): Receiver encrypts each element $y_z \in Y$ using BGV.Enc and transmits the ciphertext $\mathbf{ct} = [\![Y]\!]_p = ([\![y_1]\!]_p, [\![y_2]\!]_p, ..., [\![y_{N_Y}]\!]_p)$ to the sender.

(3) PSI.Intersection ($\mathbf{ct}, X$): Sender chooses a random number r_z for $[\![y_z]\!]_p \in \mathbf{ct}$, and computes $d_i = r_i \prod_{x \in X}([\![y_z]\!]_{p_s} - x)$. Then, the sender returns $(d_1, d_2, \ldots, d_m)$ to the receiver.

(4) PSI.Ext ($\mathbf{d}, s$): Receiver computes $Dec(d_z) = r_z \prod_{x \in X}(y_z - x)$ using BGV secret key s, and obtains y_z where $X \cap Y =$BGV.Dec$(d_z) = 0$.

4. System Models

In this section, we introduce the models proposed in our study: the network model, the adversary model, and the security model.

4.1. Network Model

The proposed system consists of four entities: a trusted authority ($\mathcal{TA}$), a data owner ($\mathcal{DO}$), a data user ($\mathcal{DU}$), and a cloud server ($\mathcal{CS}$). The system architecture is illustrated in Figure 1, and a detailed description of each entity is given as follows.

- $\mathcal{TA}$: $\mathcal{TA}$ is a trusted authority that initiates the system by generating parameters for data sharing. $\mathcal{TA}$ undertakes the task of registering both $\mathcal{DO}$ and $\mathcal{DU}$, issuing them with the necessary credentials.
- Data owner ($\mathcal{DO}$): $\mathcal{DO}$ is hospitals, clinics, or research institutions. $\mathcal{DO}$ encrypts medical data and sends them to $\mathcal{CS}$. When $\mathcal{DU}$ requests a common keyword identification query, $\mathcal{DO}$ computes an intersection set result, decryptable only by $\mathcal{DU}$ after legitimacy verification. $\mathcal{DO}$ also provides the aggregate key and relevant data class set upon $\mathcal{DU}$'s data access request.
- Data user ($\mathcal{DU}$): $\mathcal{DU}$ is a doctor, nurse, researcher, patient, etc., within a medical institution. To access data, $\mathcal{DU}$ initiates a common keyword identification query. After receiving the results, $\mathcal{DU}$ requests access to data related to the matched keyword results and then uses the aggregate key to decrypt the data obtained from CS.
- Cloud server ($\mathcal{CS}$): $\mathcal{CS}$ is an entity that stores the medical data and returns the data search results. When data are uploaded by $\mathcal{DO}$, $\mathcal{CS}$ stores the data if $\mathcal{DO}$ has the necessary legal permissions. $\mathcal{CS}$ facilitates data access to $\mathcal{DU}$ following a verification process to ascertain the legal status of $\mathcal{DU}$.

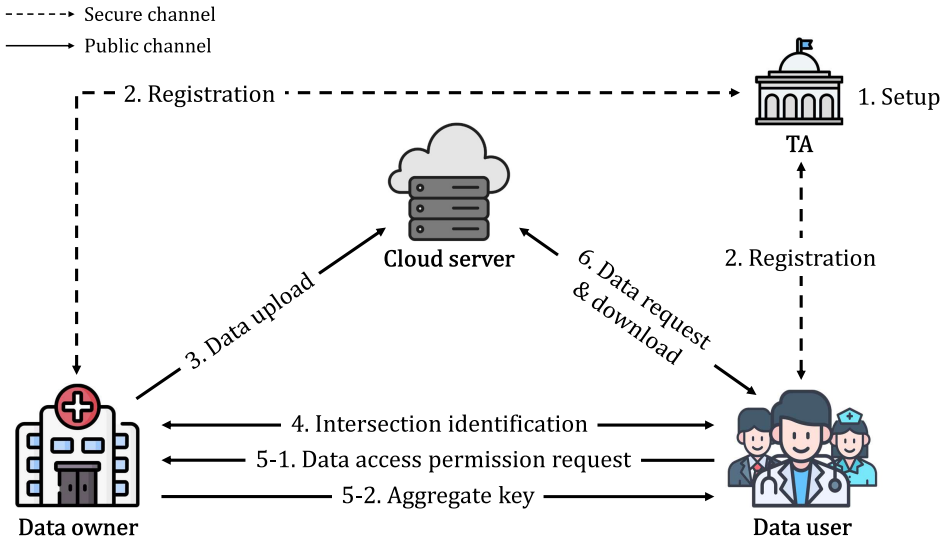

Figure 1. Network model of the proposed scheme.

The communication flows of the proposed model are summarized as follows.

(1) $\mathcal{TA}$ initializes the system parameters for authentication, intersection calculation, and data sharing.
(2) $\mathcal{TA}$ registers $\mathcal{DO}$ and $\mathcal{DU}$, storing the identity information to prevent duplicate registrations. Then, $\mathcal{TA}$ issues credentials for secure data sharing through authentication.
(3) $\mathcal{DO}$ encrypts the medical data and uploads them to $\mathcal{CS}$. $\mathcal{CS}$ then verifies the $\mathcal{DO}$'s legitimacy prior to storing the data.
(4) $\mathcal{DU}$ submits the common keyword identification query for owned information. $\mathcal{DO}$ generates and transmits the encrypted intersection results after confirming $\mathcal{DU}$'s legitimacy with $\mathcal{TA}$. $\mathcal{DU}$ verifies the received message and stores the intersection.
(5) $\mathcal{DU}$ transmits a query for data access permission to $\mathcal{DO}$ using the intersection results. Then, $\mathcal{DO}$ generates and sends the aggregate key and corresponding data class set based on the intersected keywords.

(6) $\mathcal{DU}$ requests the data from $\mathcal{CS}$ using a data class set and decrypts them using the aggregate key obtained from $\mathcal{DO}$.

4.2. Adversary Model

We adopt the Dolev–Yao (DY) model to assess the security of the proposed scheme [32], which is widely used to evaluate the security of protocols. The DY model assumes that an adversary has the ability to intercept all communications on a network, read and modify intercepted messages, and create and transmit new messages. These capabilities allow the adversary to carry out a range of attacks, such as impersonation, replay, and main-in-the-middle attacks. By analyzing the proposed scheme using the DY model, our objective is to assess its effectiveness in preventing unauthorized access, data tampering, and malicious activities planned by potential opponents.

4.3. Security Model

Aligned with the adversary model outlined in Section 4.2, the proposed scheme is designed to uphold stringent data privacy standards. Given the sensitive nature of medical data, a breach could have serious consequences. Hence, protecting the confidentiality of $\mathcal{DO}$'s information is critical to prevent unauthorized access and the subsequent leakage of sensitive data. To ensure robust data privacy, it is imperative that the ciphertext remains impervious to unauthorized decryption attempts, thereby preventing the exposure of plaintext information. In order to rigorously assess and validate the efficacy of our approach in preserving data privacy, we adopt the IND-CPA model. In this paper, we introduce the IND-CPA model game for evaluating the security posture of the proposed scheme.

Definition 1 (Data privacy)**.** *In our proposed scheme, we establish semantic security for data privacy using the IND-CPA model. The advantage of adversary $\mathcal{A}$ is quantified by $Adv_{\mathcal{A}}^{IND-CPA} = |Pr[\varkappa' = \varkappa] - \frac{1}{2}|$. The scheme achieves security against IND-CPA if, across all potential attacks, the inequality $|Pr[\varkappa' = \varkappa] - \frac{1}{2}| \leq \varepsilon$ is upheld, where ε represents a negligibly small probability.*

- *Init. $\mathcal{A}$ selects a specific set S_a from the available set $S = \{1, ..., n\}$, which it aims to exploit.*
- *Setup. The simulator $\mathcal{B}$ provides the system parameters to $\mathcal{A}$.*
- *Phase 1. For $S^* \subseteq \bar{S}_a$, $\mathcal{A}$ submits an aggregate key request query to $\mathcal{B}$. Subsequently, $\mathcal{B}$ generates and transmits the aggregate key to $\mathcal{A}$.*
- *Challenge. $\mathcal{A}$ selects two plaintexts, F_0 and F_1, of equal length from a set of possible plaintexts associated with class i_t. These plaintexts are then forwarded to $\mathcal{B}$. Thereafter, $\mathcal{B}$ obtains a random bit $\varkappa \in \{0, 1\}$ via a coin flip. Following this, $\mathcal{B}$ encrypts the selected plaintext $F_\varkappa$ and transmits the resulting ciphertext to $\mathcal{A}$.*
- *Phase 2. $\mathcal{A}$ iterates through Phase 1 for $S^* \subseteq \bar{S}_a$, encompassing classes that do not belong to S_a.*
- *Guess. $\mathcal{A}$ produces an estimate $\varkappa'$ of the true value of $\varkappa$ and communicates it to $\mathcal{B}$. If the estimate $\varkappa'$ aligns with the true value $\varkappa$, $\mathcal{A}$ is deemed successful in the game.*

5. Proposed Scheme

The proposed scheme encompasses six distinct phases: setup, registration, data upload, common keyword identification, aggregate key issuance, and data request and download. Figure 2 is the flowchart of the proposed scheme. During the setup phase, $\mathcal{TA}$ initializes the system parameters. In the registration phase, $\mathcal{TA}$ registers $\mathcal{DO}$ and $\mathcal{DU}$, providing them with the necessary credentials for data sharing. In the data upload phase, $\mathcal{DO}$ uploads the encrypted data, which are then stored by $\mathcal{CS}$ following verification of $\mathcal{DO}$'s legitimacy. The common keyword identification phase involves $\mathcal{DU}$ communicating with $\mathcal{DO}$ to acquire matching keywords. During the aggregate key issuance phase, $\mathcal{DO}$ issues an aggregate key along with the corresponding dataset based on the set of keywords requested by $\mathcal{DU}$. Finally, in the data request and download phase, $\mathcal{DU}$ can request and retrieve the data from $\mathcal{CS}$. Table 1 provides the notation utilized throughout the proposed scheme.

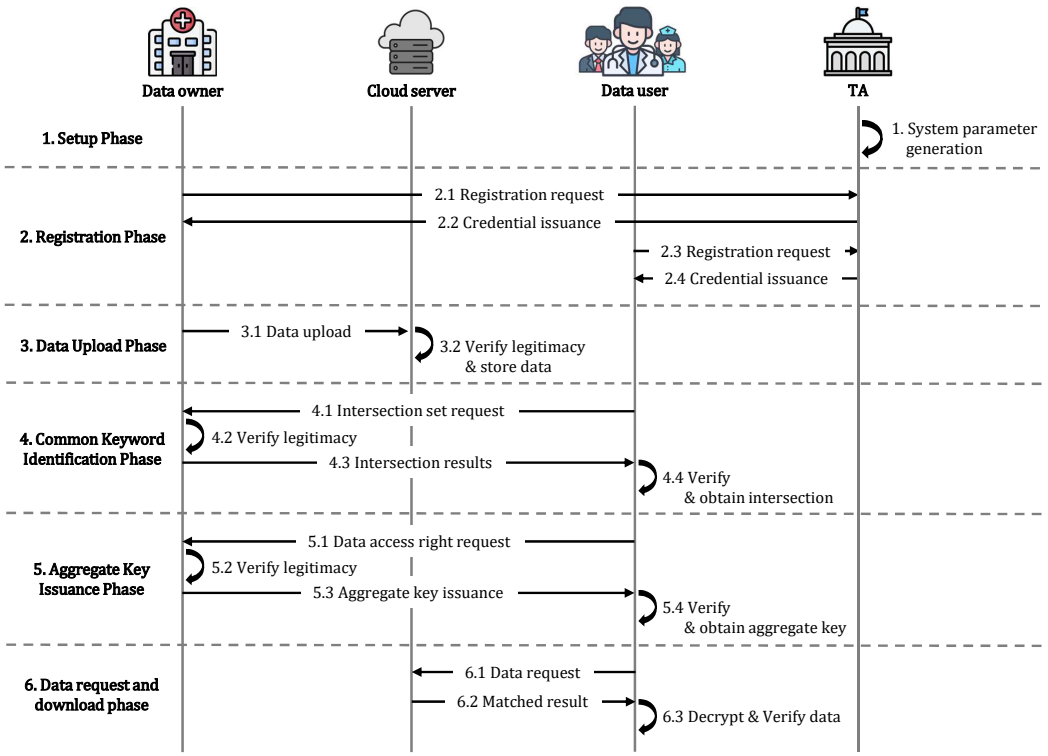

Figure 2. The overall flowchart of the proposed scheme.

Table 1. Notation.

Notation	Description
ID_o, ID_u	Identity of $\mathcal{DO}$ and $\mathcal{DU}$
(pk_{TA}, k_{TA})	$\mathcal{TA}$'s public-master key based on ECC
(pk_o, sk_o), (p_o, s_o)	$\mathcal{DO}$'s public-private key pairs based on ECC and BGV
(pk_u, sk_u), (p_u, s_u)	$\mathcal{DU}$'s public-private key pairs based on ECC and BGV
(pk_s, k_s)	$\mathcal{CS}$'s public–private key pair based on ECC
n	Maximum number of document
S	Dataset index of $\mathcal{DU}$
$\alpha, R_o, r_o, R_u, r_u, s$	Random number
e_o, e_u	Random error
$u, a_1, a_2, b_1, b_2, d_1, d_2$	Random nonce
$T_{A1}, T_{A2}, T_{B1}, T_{B2}, T_{D1}, T_{D2}$	Timestamp
$\triangle T$	Maximum transmission delay
AK	Aggregate key
$\mathcal{G}, \mathcal{G}_T$	Additive group and multiplicative group
$\hat{e}$	Bilinear map $\hat{e} : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}_T$
h	One-way hash function $h : \{0,1\}^* \rightarrow Z_p$
$\oplus$	Bitwise exclusive-or operator
$\|\|$	Concatenation operator

5.1. Setup Phase

$\mathcal{TA}$ sets a security parameter λ and chooses ciphertext modulus q, plaintext modulus t, noise distribution $\mathcal{X}$, and $R_q = \mathbb{Z}_q[X]/(X^l + 1)$. $\mathcal{TA}$ also generates the bilinear parameters $(p, \mathcal{G}, \mathcal{G}_T, \hat{e})$ and chooses a generates $P \in \mathcal{G}$ and $\alpha \in \mathbb{Z}_p$. Then, $\mathcal{TA}$ computes $P_i = \alpha^i P \in \mathcal{G}$ for $i \in \{1, \ldots, n, n+2, \ldots, 2n\}$. $\mathcal{TA}$ generates a hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}_p$ and publishes $param = \{q, t, l, \mathcal{X}, R_q, p, \mathcal{G}, \mathcal{G}_T, \hat{e}, P, \{P_i\}_{1 \leq i \leq 2n, i \neq n+1}, n, h\}$.

5.2. Registration Phase

$\mathcal{TA}$ conducts the registration of both $\mathcal{DO}$ and $\mathcal{DU}$, issuing the necessary credentials. The registration procedure is performed in a secure channel, and we present this phase only for $\mathcal{DO}$, since the registration process is identical for both $\mathcal{DO}$ and $\mathcal{DU}$.

Step 1: $\mathcal{DO}$ selects and sends ID_o to $\mathcal{TA}$.

Step 2: $\mathcal{TA}$ checks whether ID_o is registered by computing $V_o = h(ID_o||k_{TA})$. $\mathcal{TA}$ generates $r_o \in R_q$, $e_o \in \mathcal{X}$, $s_o \in \{-1, 0, 1\}^k$, and $R_o \in \mathbb{Z}_p$ and computes $p_o = (r_o \cdot s_o + t \cdot e_o, -r_o)$, $v_o = (k_{TA} + R_o) \mod p$, $d_o = R_o \cdot P$. Then, $\mathcal{TA}$ stores $V_o = h(ID_o||k_{TA})$, and sends $\{p_o, s_o, v_o, d_o\}$ to $\mathcal{DO}$.

Step 3: $\mathcal{DO}$ stores $\{p_o, s_o, v_o, d_o\}$ securely.

5.3. Data Upload Phase

For data security, $\mathcal{DO}$ computes the authentication message and encrypted data using the random nonce $s, u \in \mathbb{Z}_p$, credentials v_o, and sk_o for data F_i. Upon the message being received, $\mathcal{CS}$ stores the encrypted data $\{c_1, c_2, c_3, v_i\}$ after verifying $\mathcal{DO}$'s legal registration with $\mathcal{TA}$. The data upload process is illustrated in Figure 3, with detailed steps outlined below.

$\mathcal{DO}$	$\mathcal{CS}$
Step 1: Generates $s, u \in \mathbb{Z}_p$ Computes $U_1 = u \cdot P$, $U_2 = u \cdot pk_s$ $U_3 = u + h(U_2||v_o \cdot P) \cdot sk_o \pmod p$ Computes for document $F_i (i \in \{1, ..., n\})$ $c_1 = s \cdot P$, $c_2 = s \cdot (pk_o + P_i)$ $c_3 = F_i \cdot e(P_1, P_n)^s$ $v_i = h(F_i||ID_o)$ $C_i = (c_1||c_2||c_3||v_i) \oplus U_2$ $\xrightarrow{\{ID_o, U_1, U_3, d_o, C_i\}}$	
	Step 2: Computes $U_2^* = U_1 \cdot sk_s$ Checks $U_3 \cdot P \stackrel{?}{=} U_1 + h(U_2^*||pk_{TA} + d_o) \cdot pk_o$ Computes $(c_1||c_2||c_3||v_i) = C_i \oplus U_2$ Stores $\{c_1, c_2, c_3, v_i\}$

Figure 3. Data upload phase.

Step 1: $\mathcal{DO}$ generates a random nonce $s, u \in \mathbb{Z}_p$, and computes $U_1 = u \cdot P$, $U_2 = u \cdot pk_s$, $U_3 = u + h(U_2||v_o \cdot P) \cdot sk_o \pmod p$. $\mathcal{DO}$ also computes $c_1 = s \cdot P$, $c_2 = s \cdot (pk_o + P_i)$, $c_3 = F_i \cdot e(P_1, P_n)^s$, $v_i = h(F_i||ID_o)$, $C_i = (c_1||c_2||c_3||v_i) \oplus U_2$ for document $F_i (i \in \{1, \ldots, n\})$. Then, $\mathcal{DO}$ sends $\{ID_o, U_1, U_3, d_o, C_i\}$ to $\mathcal{CS}$.

Step 2: Upon the uploaded message, $\mathcal{CS}$ computes $U_2^* = U_1 \cdot sk_s$ and checks whether $U_3 \cdot P$ is equal to $U_1 + h(U_2^*||pk_{TA} + d_o) \cdot pk_o$. If it it correct, $\mathcal{CS}$ computes $(c_1||c_2||c_3||v_i) = C_i \oplus U_2$ and stores $\{c_1, c_2, c_3, v_i\}$.

5.4. Common Keyword Identification Phase

$\mathcal{DU}$ initiates a request to obtain common keywords related to its own data. $\mathcal{DU}$ sends the **ct** for Y with an authentication value. After receiving the query, $\mathcal{DO}$ verifies the legitimacy of $\mathcal{DU}$ through $A3$ using d_u and transmits the encrypted intersection results d_z. Subsequently, $\mathcal{DU}$ extracts the common keywords from the intersection set. Figure 4 illustrates the common keyword identification procedure, providing a detailed overview of each step.

$\mathcal{DO}$	$\mathcal{DU}$
	Step 1: Generates a_1 and T_{A1} Computes $A_1 = a_1 \cdot P$, $A_2 = a_1 \cdot pk_o$ $A_3 = a_1 + h(A_2 \| v_u \cdot P \| ID_u) \cdot sk_u \mod p$ $\mathbf{ct} = [\![Y]\!]_{p_u} = ([\![y_1]\!]_{p_u}, [\![y_2]\!]_{p_u}, ..., [\![y_m]\!]_{p_u})$ $\xleftarrow{\{T_{A1}, ID_u, d_u, A_1, A_3, \mathbf{ct}\}}$
Step 2: Checks $\|T^*_{A1} - T_{A1}\| \leq \triangle T$ Computes $A^*_2 = A_1 \cdot sk_o$ Checks $A_3 \cdot P \stackrel{?}{=} A_1 + h(A^*_2 \| pk_{TA} + d_u \| ID_u) \cdot pk_u$ Generates a_2 and T_{A2} Compute $A_4 = a_2 \cdot P$, $A_5 = a_2 \cdot pk_u$ $A_6 = a_2 + h(A_6 \| v_o \cdot P \| A^*_2 \| ID_o) \cdot sk_o \mod p$ Generates r_z for $[\![y_z]\!]_{pk_u} \in \mathbf{ct}$ Computes $d_z = r_z \prod_{x \in X}([\![y_z]\!]_{pk_u} - x)$ $\xrightarrow{\{T_{A2}, ID_o, d_o, A_4, A_6, d_z\}}$	
	Step 3: Checks $\|T^*_{A2} - T_{A2}\| \leq \triangle T$ Computes $A^*_5 = A_4 \cdot sk_u$ Checks $A_6 \cdot P \stackrel{?}{=} A_4 + h(A^*_5 \| pk_{TA} + d_o \| A_2 \| ID_o) \cdot pk_o$ Computes $Dec(d_z) = r_z \prod_{x \in X}(y_z - x)$ Inputs y in SI where $y \in X \cap Y$

Figure 4. Common keyword identification phase.

Step 1: $\mathcal{DU}$ generates a_1, T_{A1}, and computes $A_1 = a_1 \cdot P$, $A_2 = a_1 \cdot pk_o$, $A_3 = a_1 + h(A_2 \| v_u \cdot P \| ID_u) \cdot sk_u \pmod p$, $\mathbf{ct} = [\![Y]\!]_{p_u} = ([\![y_1]\!]_{p_u}, [\![y_2]\!]_{p_u}, \ldots, [\![y_m]\!]_{p_u})$. Then, $\mathcal{DU}$ sends $\{T_{A1}, ID_u, d_u, A_1, A_3, \mathbf{ct}\}$.

Step 2: After receiving the message, $\mathcal{DO}$ checks $|T^*_{A1} - T_{A1}|$ and $A_3 \cdot P \stackrel{?}{=} A_1 + h(A^*_2 \| pk_{TA} + d_u \| ID_u) \cdot pk_u$ by computing $A^*_2 = A_1 \cdot sk_o$. If it is correct, $\mathcal{DO}$ generates a_2 and T_{A2} and computes $A_4 = a_2 \cdot P$, $A_5 = a_2 \cdot pk_u$, $A_6 = a_2 + h(A_6 \| v_o \cdot P \| A^*_2 \| ID_o) \cdot sk_o \pmod p$. $\mathcal{DO}$ also generates r_z for $[\![y_z]\!]_{p_u} \in \mathbf{ct}$ and computes $d_z = r_z \prod_{x \in X}([\![y_z]\!]_{p_u} - x)$. Then, $\mathcal{DO}$ transmits $\{T_{A2}, ID_o, d_o, A_4, A_6, d_z\}$.

Step 3: $\mathcal{DU}$ checks $|T^*_{A2} - T_{A2}| \leq \triangle T$ and computes $A^*_5 = A_4 \cdot sk_u$. If $A_6 \cdot P$ is equated to $A_4 + h(A^*_5 \| pk_{TA} + d_o \| A_2 \| ID_o) \cdot pk_o$, $\mathcal{DU}$ computes $Dec(d_z) = r_z \prod_{x \in X}(y_z - x)$ using s_u, and inputs y in SI where $y \in X \cap Y$.

5.5. Aggregate Key Issuance Phase

To obtain the data access permission about intersection SI, $\mathcal{DU}$ sends the aggregate key request message $\{T_{B1}, ID_u, B_1, B_3, B_4\}$ to $\mathcal{DO}$. After confirming the validity of the $\mathcal{DU}$, $\mathcal{DO}$ provides an accessible dataset S with an aggregate key AK. Figure 5 depicts the process for aggregate key issuance, outlining the steps in detail below.

$\mathcal{DO}$	$\mathcal{DU}$
	Step 1: Generates b_1 and T_{B1} Computes $B_1 = b_1 \cdot P$, $B_2 = b_1 \cdot pk_o$ $B_3 = SI \oplus h(B_2 \| T_{B1})$ $B_4 = b_1 + h(ID_o \| ID_u \| B_2 \| SI) \cdot sk_u \mod p$ $\xleftarrow{\{T_{B1}, ID_u, B_1, B_3, B_4\}}$
Step 2: Checks $\lvert T^*_{B1} - T_{B1} \rvert \leq \triangle T$ Computes $B^*_2 = B_1 \cdot sk_o$ $SI = B_3 \oplus h(B^*_2 \| T_{B1})$ Checks $B_4 \cdot P \stackrel{?}{=} B_1 + h(ID_o \| ID_u \| B^*_2 \| SI) \cdot pk_u$ Generates b_2 and T_{B2} Compute $B_5 = b_2 \cdot P$, $B_6 = b_2 \cdot pk_u$ $AK = \sum_{j \in S} sk_o \cdot P_{n+1-j}$ $B_7 = b_2 + h(ID_o \| ID_u \| B^*_2 \| B_6 \| SI) \cdot sk_o \mod p$ $B_8 = (AK \| S) \oplus h(B^*_2 \| B_6)$ $\xrightarrow{\{T_{B2}, ID_o, B_5, B_7, B_8\}}$	
	Step 3: Checks $\lvert T^*_{B2} - T_{B2} \rvert \leq \triangle T$ Computes $B^*_6 = B_5 \cdot sk_u$ Checks $B_7 \cdot P \stackrel{?}{=} B_5 + h(ID_o \| ID_u \| B_2 \| B^*_6 \| T_{B2}) \cdot pk_o$ Computes $(AK \| S) = B_8 \oplus h(B_2 \| B^*_6)$

Figure 5. Aggregate key issuance phase.

Step 1: $\mathcal{DU}$ generates b_1, T_{B1} and computes $B_1 = b_1 \cdot P$, $B_2 = b_2 \cdot pk_o$, $B_3 = SI \oplus h(B_2 \| T_{B1})$, $B_4 = b_1 + h(ID_o \| ID_u \| B_2 \| SI) \cdot sk_u \pmod p$. Then, $\mathcal{DU}$ sends $\{T_{B1}, ID_u, B_1, B_3, B_4\}$.

Step 2: $\mathcal{DO}$ checks $\lvert T^*_{B1} - T_{B1} \rvert \leq \triangle T$, and computes $B^*_2 = B_1 \cdot sk_o$, $SI = B_3 \oplus h(B^*_2 \| T_{B1})$. If $B_4 \cdot P \stackrel{?}{=} B_1 + h(ID_o \| ID_u \| B^*_2 \| SI) \cdot pk_u$, $\mathcal{DO}$ generates b_2, T_{B2} and computes $B_5 = b_2 \cdot P$, $B_6 = b_2 \cdot pk_u$, $AK = \sum_{j \in S} sk_o \cdot P_{n+1-j}$, $B_7 = b_2 + h(ID_o \| ID_u \| B^*_2 \| B_6 \| SI) \cdot sk_o \pmod p$, $B_8 = (AK \| S) \oplus h(B^*_2 \| B_6)$. Then, $\mathcal{DO}$ transmits $\{T_{B2}, ID_o, B_5, B_7, B_8\}$.

Step 3: $\mathcal{DU}$ checks $\lvert T^*_{B2} - T_{B2} \rvert \leq \triangle T$ and computes $B^*_6 = B_5 \cdot sk_u$ for checking $B_7 \cdot P \stackrel{?}{=} B_5 + h(ID_o \| ID_u \| B_2 \| B^*_6 \| T_{B2}) \cdot pk_o$. If accurate, $\mathcal{DU}$ computes $(AK \| S) = B_8 \oplus h(B_2 \| B^*_6)$.

5.6. Data Request and Download Phase

$\mathcal{DU}$ requests data from $\mathcal{CS}$ corresponding to the dataset S received from $\mathcal{DO}$, and $\mathcal{CS}$ transmits the matched results $\{T_{D2}, c_1, v_i, PF_i\}$. $\mathcal{DU}$ then uses the aggregate key AK to decrypt the received data and obtain the document F_i. This phase is delineated in Figure 6, elucidating each sequential step below.

$\mathcal{DU}$	$\mathcal{CS}$
Step 1: Generates d_1 and T_{D1} Computes $D_1 = d_1 \cdot P$, $D_2 = d_1 \cdot pk_s$ $D_3 = S \oplus h(D_2 \| T_{D1})$ $\xrightarrow{\{T_{D1}, D_1, D_3\}}$	
	Step 2: Checks $\lvert T^*_{D1} - T_{D1} \rvert \leq \triangle T$ Computes $D^*_2 = D_1 \cdot sk_s$ $S = D_3 \oplus h(D^*_2 \| T_{D1})$ Generates T_{D2} Compute $v_1 = \sum_{j \in S, j \neq i} P_{n+1-j+i}$, $v_2 = \sum_{j \in S} P_{n+1-j}$ $PF_i = c_3 \cdot \frac{e(v_1, c_1)}{e(v_2, c_2)}$ $\xleftarrow{\{T_{D2}, c_1, v_i, PF_i\}}$
Step 3: Checks $\lvert T^*_{D2} - T_{D2} \rvert \leq \triangle T$ Computes $F^*_i = PF_i \cdot e(AK, c_1)$ Checks $v_i \stackrel{?}{=} h(F^*_i \| ID_i)$	

Figure 6. Data request and download phase.

Step 1: $\mathcal{DU}$ generates d_1, T_{D1}, computes $D_1 = d_1 \cdot P$, $D_2 = d_1 \cdot pk_s$, $D_3 = S \oplus h(D_2 \| T_{D1})$, and sends $\{T_{D1}, D_1, D_3\}$.

Step 2: According to the received message, $\mathcal{CS}$ checks $\lvert T^*_{D1} - T_{D1} \rvert \leq \triangle T$ and computes $D^*_2 = D_1 \cdot sk_s$, $S = D_3 \oplus h(D^*_2 \| T_{D1})$. For S, $\mathcal{CS}$ generates T_{D2} and computes $v_1 = \sum_{j \in S, j \neq i} P_{n+1-j+i}$, $v_2 = \sum_{j \in S} P_{n+1-j}$, $PF_i = c_3 \cdot \frac{e(v_1, c_1)}{e(v_2, c_2)}$. Then, $\mathcal{CS}$ sends $\{T_{D2}, c_1, v_i, PF_i\}$.

Step 3: $\mathcal{DU}$ checks $\lvert T^*_{D2} - T_{D2} \rvert \leq \triangle T$ and obtains data F_i by computing $F^*_i = PF_i \cdot e(AK, c_1)$. To verify the data, $\mathcal{DU}$ checks whether v_i is equal to $h(F^*_i \| ID_i)$.

Correctness:

$$
\begin{aligned}
F_i &= PF_i \cdot e(AK, c_1) \\
&= c_3 \cdot \frac{e(v_1, c_1)}{e(v_2, c_2)} \cdot e(AK, c_1) \\
&= c_3 \cdot \frac{e(\sum_{j \in S, j \neq i} P_{n+1-j+i}, s \cdot P)}{e(\sum_{j \in S} P_{n+1-j}, s \cdot (pk_o + P_i))} \cdot e(\sum_{j \in S} sk_o \cdot P_{n+1-j}, s \cdot P) \\
&= c_3 \cdot \frac{e(\sum_{j \in S, j \neq i} P_{n+1-j+i}, s \cdot P)}{e(\sum_{j \in S} P_{n+1-j}, s \cdot pk_o) \cdot e(\sum_{j \in S} P_{n+1-j}, s \cdot P_i)} \cdot e(\sum_{j \in S} sk_o \cdot P_{n+1-j}, s \cdot P) \\
&= c_3 \cdot \frac{e(\sum_{j \in S} sk_o \cdot P_{n+1-j}, s \cdot P)}{e(\sum_{j \in S} P_{n+1-j}, s \cdot pk_o) \cdot e(\alpha^{n+1} P, s \cdot P)} \\
&= F_i \cdot \frac{e(P_1, P_n)^s}{e(\alpha^{n+1} P, s \cdot P)} \\
&= F_i \cdot \frac{e(P_1, P_n)^s}{e(P_1, P_n)^s} \\
&= F_i
\end{aligned}
$$

6. Security Analysis

We conduct a comprehensive security analysis to prove the resilience of our proposed scheme. Our assessment encompasses potential threats, ranging from informal attack scenarios to formal analysis. In the informal security analysis, we evaluate whether the proposed scheme meets essential security requirements, including resilience against impersonation, replay, and denial-of-service attacks, as well as ensuring mutual authentication and data privacy. For the formal analysis, we use IND-CPA to verify the robustness of data privacy protections. We employ BAN logic to confirm the guarantee of mutual authentication and utilize the Scyther tool to validate the security of the proposed scheme against potential vulnerabilities, focusing on common keyword confirmation and integrated key issuance.

6.1. Informal Security Analysis

We perform a security evaluation to estimate the robustness of the proposed scheme against various threats that can occur in a medical data sharing environment. We also verify that mutual authentication between entities is provided during communication. We consider that an adversary endeavors security breaches founded on the suppositions delineated in Section 4.2.

6.1.1. Impersonation Attack

$\mathcal{A}$ endeavors to impersonate $\mathcal{DU}$ in an effort to intercept the transmitted messages between $\mathcal{DO}$, $\mathcal{DU}$, and $\mathcal{CS}$, aiming to obtain sensitive data. $\mathcal{A}$ initiates the transmission of a data request message, denoted as $\{T_{D1}, D_1, D_3\}$, to $\mathcal{CS}$ as outlined in Section 5.6. However, $\mathcal{A}$ cannot compute the message without the dataset S. $\mathcal{A}$ also endeavors to extract data from an intercepted message $\{T_{D2}, c_1, v_i, PF_i\}$, but it is impossible without an aggregate key AK. In an attempt to acquire the AK and S of $\mathcal{DU}$, $\mathcal{A}$ endeavors to transmit $\{T_{B1}, ID_u, B_1, B_3, B_4\}$ in Section 5.5. However, $\mathcal{A}$ faces insurmountable barriers as it lacks crucial information including $\mathcal{DU}$'s secret key sk_u, a random nonce b_1, the common keyword set SI, and the identity of the data owner ID_o. Even if $\mathcal{A}$ tries to obtain AK and S from $\{T_{B2}, ID_o, B_5, B_7, B_8\}$, it is impossible because $\mathcal{A}$ needs sk_u. Furthermore, attempts to access SI and ID_o for the desired data detailed in Section 5.4 are futile due to $\mathcal{A}$'s lack of knowledge about sk_u and a_1. Consequently, the security of the proposed system against impersonation attacks is affirmed.

6.1.2. Replay and Man-in-the-Middle (MITM) Attack

$\mathcal{A}$ tries to resend the common keyword confirmation message $\{T_{A1}, ID_u, d_u, A_1, A_3, \mathbf{ct}\}$, data access permission request message $\{T_{B1}, ID_u, B_1, B_3, B_4\}$, and data request message $\{T_{D1}, D_1, D_3\}$ with the purpose of obtaining data. However, these messages consist of timestamps T_{A1}, T_{B1}, T_{D1}, and random nonces a_1, b_1, d_1, and each entity that receives the message checks its freshness. Even if $\mathcal{A}$ retransmits a previous message, entities can distinguish it as a malicious message. $\mathcal{A}$ also intercepts and attempts to modify the messages, but it is impossible without the knowledge of $sk_u, sk_o, a_1, a_2, b_1, b_2, d_1, d_2$. Hence, our scheme resists the replay and MITM attacks.

6.1.3. Denial of Services (DoS) Attack

$\mathcal{A}$ seeks to disrupt availability by inundating $\mathcal{CS}$ with an overwhelming volume of messages, thereby overloading its capacity or halting data sharing services altogether. During such an attack, $\mathcal{A}$ ruthlessly transmits data upload messages $\{ID_o, U_1, U_3, d_o, C_i\}$ and data request messages $\{T_{D1}, D_1, D_3\}$ to $\mathcal{CS}$. However, $\mathcal{CS}$ effectively mitigates this threat by scrutinizing the timestamps of incoming messages and promptly interrupting any deemed invalid. Consequently, the proposed system robustly defends against DoS attacks, ensuring uninterrupted service availability.

6.1.4. Mutual Authentication

In the proposed scheme, legitimacy is verified between communication entities to ensure secure medical data sharing. In Section 5.4, when $\mathcal{DU}$ requests the common keyword results from $\mathcal{DO}$ through $\{T_{A1}, ID_u, d_u, A_1, A_3, \mathbf{ct}\}$, upon receiving the message, $\mathcal{DO}$ checks whether $\mathcal{DU}$ has been legitimately registered in $\mathcal{TA}$ via $A_3 \cdot P \stackrel{?}{=} A_1 + h(A_2^*||pk_{TA} + d_u||ID_u) \cdot pk_u$. If this verification is successful, $\mathcal{DO}$ sends $\{T_{B2}, ID_o, B_5, B_7, B_8\}$ along with the common keyword identification function $d_z = r_z \prod_{x \in X}([\![y_z]\!]_{pk_u} - x)$, which can be decrypted by $\mathcal{DU}$. $\mathcal{DU}$ then verifies the correctness of the message sent by the $\mathcal{DO}$, who has legally registered with $\mathcal{TA}$, via $A_6 \cdot P \stackrel{?}{=} A_4 + h(A_5^*||pk_{TA} + d_o||A_2||ID_o) \cdot pk_o$. Mutual authentication is performed in the same way at other phases. Therefore, the proposed scheme ensures mutual authentication.

6.1.5. Data Verification

Upon receiving the results of the data request query $\{T_{D2}, c_1, v_i, PF_i\}$, $\mathcal{DU}$ proceeds with data verification. This involves computing $F_i^* = PF_i \cdot e(AK, c_1)$ and subsequently checking whether $v_i \stackrel{?}{=} h(F_i^*||ID_i)$. This verification process ensures that the received data have not been tampered with. By adding an additional layer of security, the proposed scheme reinforces the integrity of transmitted data. Therefore, it not only facilitates secure medical data sharing but also prioritizes data integrity, mitigating the risk of unauthorized modifications.

6.2. Semantic Security

In Theorem 1, we show that the proposed scheme provides IND-CPA security.

Theorem 1. *Given a probabilistic polynomial-time adversary $\mathcal{A}$ with a non-negligible advantage ε, $\mathcal{A}$ is capable of tackling the formidable assumption problem with a gain of $\frac{\varepsilon}{2}$.*

Proof of Theorem 1. Let $\mathcal{A}$ be an entity capable of compromising the proposed scheme with an advantage of ε. In response, we introduce $\mathcal{B}$ to engage in the DBDH game, achieving an advantage of $\varepsilon/2$. The challenger $\mathcal{C}$ selects a generator $P \in \mathcal{G}$ and four random values $a, b, c, d \in \mathbb{Z}_p$. $\mathcal{C}$ then randomly determines a value $\varkappa \in \{0, 1\}$ and shares it with $\mathcal{B}$. If $\varkappa = 0$, $\mathcal{C}$ computes $\mathcal{V} = \hat{e}(P, P)^{abc}$, resulting in the tuple $(aP, bP, cP, \hat{e}(P, P)^{abc})$. Otherwise, if $\varkappa = 1$, $\mathcal{C}$ computes $\mathcal{V} = \hat{e}(P, P)^{d}$, resulting in the tuple $(aP, bP, cP, \hat{e}(P, P)^{d})$.

Init. $\mathcal{B}$ employs $\mathcal{A}$ to produce a distinct subset S_a from the existing set $S = \{1, \ldots, n\}$, which $\mathcal{A}$ aims to focus on. Afterward, $\mathcal{A}$ delivers this selected set to $\mathcal{B}$.

Setup. $\mathcal{B}$ formulates the public parameters $\{P_i\}_{1 \le i \le 2n, i \ne n+1}$, where $\alpha^1 = a$ and $\alpha^n = b$. Then, $\mathcal{B}$ disseminates these parameters to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ submits an AK query for $S^* \subseteq \bar{S}_a$, and $\mathcal{B}$ responds to $\mathcal{A}$ by calculating AK as $AK = \sum_{j \in S^*} sk_o \cdot P_{n+1-j}$.

Challenge. $\mathcal{A}$ submits two plaintexts of equal length, denoted as F_0 and F_1, along with S^* to $\mathcal{B}$. $\mathcal{B}$ randomly flips a coin to determine $\varkappa \in \{0, 1\}$. If $\varkappa = 0$ and $\mathcal{V} = \hat{e}(P, P)^{abc}$, we set $s = c$, then $\hat{e}(P, P)^{abc} = \hat{e}(P, P)^{ab \cdot s} = \hat{e}(aP, bP)^s = \hat{e}(P_1, P_n)^s$ and $c_3 = F_\varkappa \cdot \hat{e}(g, g)^{abc}$ is computed. Otherwise, if $\varkappa = 1$, then $\mathcal{V} = \hat{e}(P, P)^d$ and $c_3 = F_\varkappa \cdot \hat{e}(P, P)^d$. $\mathcal{B}$ also calculates $c_1 = s \cdot P$, $c_2 = s \cdot (pk_o + P_i)$, and sends $\{c_1, c_2, c_3\}$ to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ repeats Phase 1 to obtain AK within $S^* \subseteq \bar{S}_a$.

Guess. $\mathcal{A}$ hypothesizes $\varkappa'$ to guess $\varkappa$. If $\varkappa' = \varkappa$, $\mathcal{B}$ returns 0, indicating $\mathcal{V} = \hat{e}(P, P)^{abc}$, and $\mathcal{A}$, with an advantage of ε, can practically obtain the ciphertext, resulting in a probability $Pr[\varkappa' = \varkappa | \mathcal{V} = \hat{e}(P, P)^{abc}] = \frac{1}{2} + \varepsilon$. If $\varkappa' \ne \varkappa$, $\mathcal{B}$ returns 1, indicating $\mathcal{V} = \hat{e}(P, P)^d$, and $\mathcal{A}$ receives an invalid ciphertext. Therefore, by correctly guessing $\varkappa'$, $\mathcal{A}$ gains no significant advantage, and the probability of success in the game is $Pr[\varkappa \ne \varkappa | \mathcal{V} = \hat{e}(P, P)^d] = \frac{1}{2}$. The probability Pr of a successful game can be calculated as

$$
\begin{aligned}
Pr &= \frac{1}{2}Pr[\mathcal{A}(aP, bP, cP, \hat{e}(P,P)^{abc}) = 1] + \frac{1}{2}Pr[\mathcal{A}(aP, bP, cP, \hat{e}(P,P)^{d}) = 1] - \frac{1}{2} \\
&= \frac{1}{2}Pr[\varkappa' = \varkappa | \mathcal{V} = \hat{e}(P,P)^{abc}] + \frac{1}{2}Pr[\varkappa' \neq \varkappa | \mathcal{V} = \hat{e}(P,P)^{d}] - \frac{1}{2} \\
&= \frac{1}{2} \times (\frac{1}{2} + \varepsilon) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} = \frac{\varepsilon}{2}
\end{aligned}
$$

Hence, the proposed scheme provides IND-CPA security. □

6.3. Formal Security Analysis Using BAN Logic

In the proposed scheme, $\mathcal{DO}$ and $\mathcal{DU}$ perform mutual authentication in Section 5.4 to prove that they are entities correctly registered in TA before performing an intersection. To demonstrate the mutual authentication of our scheme, we utilize a widely recognized formal verification technique called BAN logic [13]. Many researchers have affirmed the mutual authentication of their approaches using BAN logic [33,34]. To incorporate our approach with BAN logic, we provide the following notations and descriptions. Table 2 is the notation used in BAN logic.

Table 2. BAN logic notation.

Notation	Description
$\mathcal{Q} \mid\equiv \mathcal{M}$	$\mathcal{O}$ believes statement $\mathcal{M}$
$\#\mathcal{M}$	Statement $\mathcal{M}$ is fresh
$\mathcal{Q} \triangleleft \mathcal{M}$	$\mathcal{Q}$ receives statement $\mathcal{M}$
$\mathcal{Q} \mid\sim \mathcal{M}$	$\mathcal{Q}$ once said $\mathcal{M}$
$\mathcal{Q} \Rightarrow \mathcal{M}$	$\mathcal{Q}$ controls statement $\mathcal{M}$
$\langle \mathcal{M} \rangle_{\mathcal{L}}$	Statement $\mathcal{M}$ is combined with secret statement $\mathcal{L}$
$\mathcal{Q} \overset{\mathcal{L}}{\rightleftharpoons} \mathcal{K}$	$\mathcal{L}$ is a secret known only to $\mathcal{Q}$ and $\mathcal{K}$

6.3.1. Rules

The rules employed for analyzing the security scheme in BAN logic are outlined below.

- Message mearning rule (MMR):

$$\frac{\mathcal{Q} \mid\equiv \mathcal{Q} \overset{\mathcal{L}}{\rightleftharpoons} \mathcal{K}, \mathcal{Q} \triangleleft \langle \mathcal{M} \rangle_{\mathcal{L}}}{\mathcal{Q} \mid\equiv \mathcal{K} \mid\sim \mathcal{M}}$$

- Freshness rule (FR):

$$\frac{\mathcal{Q} \mid\equiv \#(\mathcal{M})}{\mathcal{Q} \mid\equiv \#(\mathcal{M}, \mathcal{S})}$$

- Nonce verification rule (NVR):

$$\frac{\mathcal{Q} \mid\equiv \#(\mathcal{M}), \mathcal{Q} \mid\equiv \mathcal{K} \mid\sim \mathcal{M}}{\mathcal{Q} \mid\equiv \mathcal{K} \mid\equiv \mathcal{M}}$$

- Jurisdiction rule (JR):

$$\frac{\mathcal{Q} \mid\equiv \mathcal{K} \mid\Rightarrow \mathcal{M}, \mathcal{Q} \mid\equiv \mathcal{K} \mid\equiv \mathcal{M}}{\mathcal{Q} \mid\equiv \mathcal{M}}$$

- Belief rule (BR):

$$\frac{\mathcal{Q} \mid\equiv \mathcal{K} \mid\equiv (\mathcal{M}, \mathcal{S})}{\mathcal{Q} \mid\equiv \mathcal{K} \mid\equiv \mathcal{M}}$$

6.3.2. Goals

The goals for checking the adequacy of the authentication properties of the proposed scheme are defined as follows.

Goal 1: $\mathcal{DO} \mid\equiv A_3$
Goal 2: $\mathcal{DO} \mid\equiv \mathcal{DU} \mid\equiv A_3$
Goal 3: $\mathcal{DU} \mid\equiv A_6$
Goal 4: $\mathcal{DU} \mid\equiv \mathcal{DO} \mid\equiv A_6$

6.3.3. Assumptions

The assumptions driving the analysis are presented as follows.

A_1**:** $\mathcal{DO} \mid\equiv \mathcal{DO} \overset{A_2}{\rightleftharpoons} DU$
A_2**:** $\mathcal{DO} \mid\equiv \#(T_{A1})$
A_3**:** $\mathcal{DO} \mid\equiv \mathcal{DU} \mid\Rightarrow A_3$
A_4**:** $\mathcal{DU} \mid\equiv \mathcal{DO} \overset{A_5}{\rightleftharpoons} DU$
A_5**:** $\mathcal{DU} \mid\equiv \#(T_{A2})$
A_6**:** $\mathcal{DU} \mid\equiv \mathcal{DO} \mid\Rightarrow A_6$

6.3.4. Idealized Forms

The idealized forms for messages exchanged among communication entities are outlined below.

M_1**:** $\mathcal{DU} \rightarrow \mathcal{DO} : \langle T_{A1}, ID_u, v_u, A_3 \rangle_{A_2}$
M_2**:** $\mathcal{DO} \rightarrow \mathcal{DU} : \langle T_{A2}, ID_o, v_o, A_6 \rangle_{A_5}$

6.3.5. Proof

In accordance with the provided rules, idealized forms, and assumptions, the analytical process aimed at achieving the goals of the proposed scheme is outlined as follows.

Step 1: S_1 can be obtained from M_1.

$$S_1 : \mathcal{DO} \triangleleft \langle T_{A1}, ID_u, v_u, A_3 \rangle_{A_2}$$

Step 2: S_2 can be obtained by applying the MMR with A_1.

$$S_2 : \mathcal{DO} \mid\equiv \mathcal{DU} \mid\sim (T_{A1}, ID_u, v_u, A_3)$$

Step 3: S_3 can be obtained by applying the FR with S_2 and A_2.

$$S_3 : \mathcal{DO} \mid\equiv \#(T_{A1}, ID_u, v_u, A_3)$$

Step 4: S_4 can be obtained by applying the NVR with S_2 and S_3.

$$S_4 : \mathcal{DO} \mid\equiv \mathcal{DU} \mid\equiv (T_{A1}, ID_u, v_u, A_3)$$

Step 5: S_5 can be obtained by applying the BR with S_4.

$$S_5 : \mathcal{DO} \mid\equiv \mathcal{DU} \mid\equiv A_3 \quad \text{(Goal 2)}$$

Step 6: S_6 can be obtained by applying the JR with S_5 and A_3.

$$S_6 : \mathcal{DO} \mid\equiv A_3 \quad \text{(Goal 1)}$$

Step 7: S_7 can be obtained from M_2.

$$S_7 : \mathcal{DU} \triangleleft \langle T_{A2}, ID_o, v_o, A_6 \rangle_{A_5}$$

Step 8: S_8 can be obtained by applying the MMR with A_4.

$$S_8 : \mathcal{DU} \mid\equiv \mathcal{DO} \mid\sim (T_{A2}, ID_o, v_o, A_6)$$

Step 9: S_9 can be obtained by applying the FR with S_8 and A_5.

$$S_9 : \mathcal{DU} \mid\equiv \#(T_{A2}, ID_o, v_o, A_6)$$

Step 10: S_{10} can be obtained by applying the NVR with S_8 and S_9.

$$S_{10} : \mathcal{DU} \mid\equiv \mathcal{DO} \mid\equiv (T_{A2}, ID_o, v_o, A_6)$$

Step 11: S_{11} can be obtained by applying the BR with S_{10}.

$$S_5 : \mathcal{DU} \mid\equiv \mathcal{DO} \mid\equiv A_6 \quad \text{(Goal 4)}$$

Step 12: S_{12} can be obtained by applying the JR with S_{11} and A_6.

$$S_{12} : \mathcal{DU} \mid\equiv A_6 \quad \text{(Goal 3)}$$

Therefore, all goals are accomplished, and the proposed scheme delivers mutual authentication.

6.4. Scyther Tool

We utilize the Scyther tool for the formal security analysis of the proposed scheme. Scyther is a push-button tool designed for the verification and analysis of the security protocol [12]. It offers extensive verification capabilities, ensuring termination while verifying the correctness of the scheme across an unlimited number of sessions. Scyther also provides features for model checking and multi-protocol analysis, complemented by a Python-based graphical user interface. These functionalities streamline the process of identifying and addressing security vulnerabilities within systems by users. Scyther delineates roles and events, representing message transmission and reception, based on the Security Protocol Description Language (SPDL). The Scyther command-line tool evaluates the security of a proposed protocol by scrutinizing the various claim events described in Table 3. Upon completion of the simulation, the result window confirms the security robustness of the proposed protocol. A status of "OK" in the "Status" tab, along with "No attacks" in the "Comment" tab, assures the security of the authentication process. Figure 7 presents the simulation result of the proposed scheme, showing the "OK" status and "No attacks" comments in all claim events. Therefore, we ensure the robustness of the security measures implemented.

Table 3. Scyther tool claim events.

Claim Event	Description
Secrecy	Confirms that sensitive information remains confidential during communication
Alive	Verifies active participation of communicating parties
Weakagree	Checks whether the communicating participant is active user or not
Niagree	Ensures an implicit agreement between communicating participants
Nisynch	Ensures messages are exchanged in the proper order from authorized participants

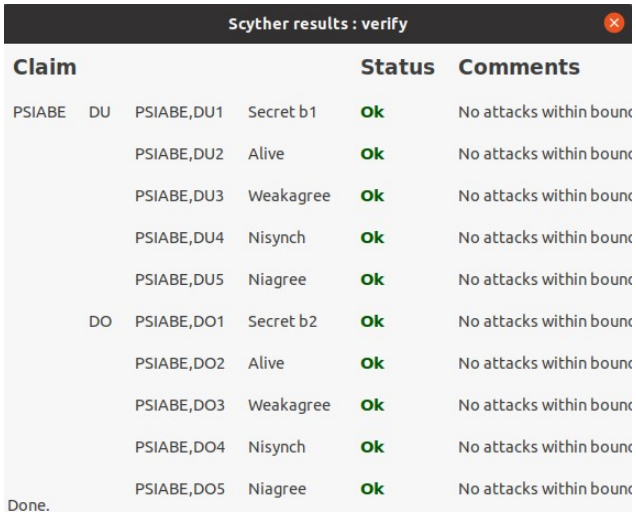

Figure 7. Scyther results.

7. Comparative Analysis

We perform an evaluative comparison regarding the security and efficiency metrics of our approach against pertinent existing frameworks.

7.1. Security Features

To ensure data confidentiality and privacy in a medical information system, it is imperative that only authorized data users should be granted access to the data, with no information related to data being disclosed. Achieving this necessitates the implementation of robust security measures to thwart unauthorized access attempts, secure message exchanges between entities, and grant data access only following thorough verification via mutual authentication. Moreover, it is crucial to maintain data integrity by verifying the authenticity of the information accessed by data users. In this context, we evaluate the security features of our proposed scheme against existing related schemes to determine its effectiveness in thwarting potential threats such as impersonation, replay, MITM, and DoS attacks. In addition, our evaluation focuses on verifying the robustness of mutual authentication, data integrity verification, and the prevention of data privacy leaks. Table 4 delineates the analysis results, comparing our proposed scheme with existing ones in terms of their capability to address the aforementioned security concerns. Based on our findings, existing studies lack robustness against DoS attacks, do not adequately consider MITM attacks, and lack essential features such as mutual authentication, data verification, or data privacy. In contrast, our proposed scheme meets the security requirements for secure data sharing within medical information systems.

Table 4. Security features.

Security Features	[18]	[19]	[22]	[24]	Ours
Replay attack	○	○	—	○	○
MITM attack	○	○	—	○	○
Impersonation attack	○	○	○	○	○
DoS attack	○	×	—	×	○
Mutual authentication	○	○	×	×	○
Data verification	○	×	×	×	○
Data Privacy	×	×	×	×	○

○: Support/resist the security features; ×: Does not support/resist the security features; —: Not applicable.

7.2. Computational Costs

We investigated the execution time of cryptographic operations on personal computers (PCs) using MIRACL [14], a software tool designed to facilitate the practical implementation

of cryptographic techniques and algorithms. The PC's specifications are as follows: Ubuntu 20.04.6 LTS operating system, 16 GB of RAM, and an Intel Core i5-10400 processor operating at 2.90 GHz (64-bit CPU). To ensure the accuracy of the measurements, we calculated the average duration of 100 iterations for each cryptographic operation, and the results are in Table 5.

Table 5. Execution time of each cryptographic operation.

Notation	Description	Execution Time
$T_{bp}^{\mathcal{G}_m}$	Bilinear pairing $\hat{e}: \mathcal{G}_m \times \mathcal{G}_m \rightarrow \mathcal{G}_{mT}$ ($\mathcal{G}_m$: multiplicative group)	4.717 ms
$T_e^{\mathcal{G}_{mT}}$	Exponentiation in $\mathcal{G}_{mT}$	1.990 ms
$T_m^{\mathcal{G}_{mT}}$	Multiplication/Division in $\mathcal{G}_{mT}$	0.032 ms
$T_m^{\mathcal{G}_m}$	Multiplication in $\mathcal{G}_m$	0.323 ms
$T_a^{\mathcal{G}_m}$	Point addition in $\mathcal{G}_m$	0.013 ms
$T_{bp}^{\mathcal{G}_a}$	Bilinear pairing $\hat{e}: \mathcal{G}_a \times \mathcal{G}_a \rightarrow \mathcal{G}_{aT}$ ($\mathcal{G}_a$: additive group)	3.023 ms
$T_e^{\mathcal{G}_{aT}}$	Exponentiation in $\mathcal{G}_{aT}$	0.341 ms
$T_m^{\mathcal{G}_{aT}}$	Multiplication/Division in $\mathcal{G}_{aT}$	0.027 ms
$T_m^{\mathcal{G}_a}$	Multiplication in $\mathcal{G}_a$	0.172 ms
$T_a^{\mathcal{G}_a}$	Point addition in $\mathcal{G}_a$	0.003 ms
$T_m^{\mathbb{Z}}$	Multiplication in $\mathbb{Z}_p$	0.006 ms
$T_a^{\mathbb{Z}}$	Addition in $\mathbb{Z}_p$	0.005 ms
T_e	Modular exponentiation	0.094 ms
T_s	Symmetric key encryption/decryption	0.001 ms
T_h	SHA-256 hash function	0.001 ms

We analyzed the message execution time on the public channel. We remain consistent in treating the keywords and attributes discussed in each paper as 1 to compare the computational cost with increasing data volume, denoted by κ. The comparison results are laid out in Table 6. As depicted in Figure 8, our proposed scheme demonstrates the lowest execution times with increasing data volume. Within medical information systems, the seamless exchange of vast datasets between data owners and users is critical for driving research, advancing medical technologies, and enhancing service delivery. Therefore, our scheme is not only efficient but also well suited for real-world medical information systems.

Table 6. Execution time comparison.

Scheme	Execution Times (ms)
[18]	$3T_{bp}^{\mathcal{G}_m} + T_e^{\mathcal{G}_{mT}} + 16T_m^{\mathcal{G}_m} + 6T_m^{\mathbb{Z}} + 4T_a^{\mathcal{G}_m} + 22T_h + \kappa(2T_{bp}^{\mathcal{G}_m} + T_m^{\mathcal{G}_{mT}} + 2T_a^{\mathcal{G}_m} + T_h) \approx 9.493\kappa + 24.419$
[19]	$8T_{bp}^{\mathcal{G}_m} + 8T_e^{\mathcal{G}_{mT}} + 4T_m^{\mathcal{G}_{mT}} + 11T_m^{\mathcal{G}_m} + 2T_m^{\mathbb{Z}} + 6T_a^{\mathcal{G}_m} + 5T_h + \kappa(2T_{bp}^{\mathcal{G}_m} + T_m^{\mathcal{G}_{mT}} + 4T_a^{\mathcal{G}_m} + T_h) \approx 9.519\kappa + 57.432$
[22]	$6T_m^{\mathcal{G}_m} + 7T_m^{\mathbb{Z}} + 4T_a^{\mathbb{Z}} + \kappa(4T_{bp}^{\mathcal{G}_m} + 2T_m^{\mathcal{G}_{mT}} + 4T_m^{\mathcal{G}_m} + 9T_m^{\mathbb{Z}} + T_a^{\mathcal{G}_m} + T_a^{\mathbb{Z}}) \approx 20.296\kappa + 28.346$
[24]	$6T_{bp}^{\mathcal{G}_m} + T_e^{\mathcal{G}_{mT}} + 6T_m^{\mathcal{G}_m} + 5T_m^{\mathbb{Z}} + 2T_a^{\mathcal{G}_m} + T_a^{\mathbb{Z}} + T_e + T_s + 4T_h + \kappa(7T_{bp}^{\mathcal{G}_m} + 3T_m^{\mathcal{G}_{mT}} + 4T_m^{\mathcal{G}_m} + 3T_m^{\mathbb{Z}} + 2T_a^{\mathcal{G}_m} + 2T_e + T_s + 2T_h) \approx 34.642\kappa + 32.39$
Ours	$T_{bp}^{\mathcal{G}_a} + T_e^{\mathcal{G}_{aT}} + 38T_m^{\mathcal{G}_a} + 11T_m^{\mathbb{Z}} + 9T_a^{\mathcal{G}_a} + 10T_a^{\mathbb{Z}} + 16T_h + \kappa(3T_{bp}^{\mathcal{G}_a} + T_m^{\mathcal{G}_{aT}} + 2T_m^{\mathcal{G}_a} + T_h) \approx 9.441\kappa + 11.708$

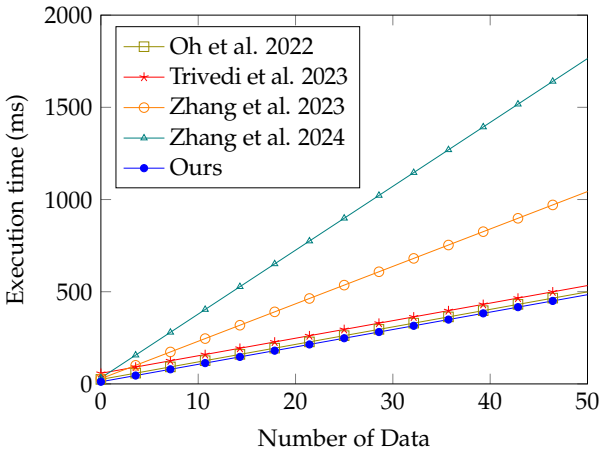

Figure 8. Comparison of the execution times with the number of data [18,19,22,24].

7.3. Time Complexity Comparison

We conduct a comparative analysis of time complexity concerning computation and communication costs in relation to existing studies. Regarding computation costs, our analysis encompasses encryption, request, and verification. Encryption involves the process of encrypting data from the owner. Request refers to the process of a user desiring access to specific data, while verification entails confirming the accuracy and reliability of received encrypted data. Regarding communication costs, we define the access key as the decryption key, the request as the user's data access query, and the ciphertext as the encrypted data received from the cloud server. As illustrated in Table 7, our comparison demonstrates significantly lower time complexity for both computation and communication compared to existing methods. Thus, our proposed approach offers enhanced efficiency and performance, rendering it more suitable for medical data sharing systems.

Table 7. Time complexity comparison.

Scheme	Computation Cost			Communication Cost		
	Encryption	Request	Verification	Access Key	Request	Ciphertext
[18]	$O(\|KW\|E)$	$O(\|Q\|E)$	$O(\|S\|P)$	$O(1)$	$O(1)$	$O(1)$
[19]	$O(\|KW\|P)$	$O(\|Q\|M)$	$O(\|Q\|P)$	$O(1)$	$O(\|Q\|)$	$O(\|Q\|)$
[22]	$O(\|A\|E)$	NA	$O(\|A\|P)$	$O(\|A\|)$	NA	$O(1)$
[24]	$O(\|KW\|P)$	$O(\|Q\|H)$	$O(\|Q\|P)$	$O(\|A\|)$	$O(\|Q\|)$	$O(\|Q\|)$
Ours	$O(1)$	$O(1)$	$O(\|S\|P)$	$O(1)$	$O(1)$	$O(1)$

$|KW|$: the number of keywords with the ciphertext; $|A|$: the number of attributes in access policy; $|S|$: the number of data; $|Q|$: the number of keyword in query set; P: pairing; M: multi-scalar multiplication; H: hash; E: exponentiation; NA: not appliable.

8. Conclusions

We have proposed a secure and privacy-preserving data sharing scheme designed for medical information systems. This scheme leverages KAE to facilitate secure and flexible data sharing between data owners and users and incorporates PSI techniques to achieve a balance between data privacy and flexible sharing. The security of our proposed scheme was rigorously evaluated through both informal and formal security analyses. Through the use of BAN logic, we ensured the scheme supports mutual authentication, while semantic secrecy was employed to prove data privacy. Additionally, the robustness of our scheme was validated using the Scyther tool, confirming its resilience against potential security threats. Our assessment extended to a comparative analysis of the security properties, execution times, and complexities, contrasting our scheme with existing methodologies. This comparison highlighted the improved security and efficiency metrics of our scheme. In conclusion, the proposed data sharing scheme not only meets the stringent security and privacy requirements of medical information systems but also exhibits superior

performance and flexibility. However, as our system employs homomorphic encryption in determining the intersection of private sets, there may be a computational burden on each entity. Hence, we intend to pursue future research aimed at identifying intersections using a lighter methodology. In addition, since there is a possibility of advanced security risks due to the development of quantum computing technology, we will consider studies to improve the resilience to these security threats after the proposed method is employed.

Author Contributions: Conceptualization, J.O.; methodology, J.O. and S.S.; software, D.K.; validation, Y.P. (Yohan Park) and M.K.; formal analysis, J.O. and S.S.; investigation, J.O. and M.K.; writing—original draft preparation, J.O.; writing—review and editing, S.S. and Y.P. (Yohan Park); supervision, Y.P. (Youngho Park); funding acquisition, Y.P. (Youngho Park). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) and funded by the Ministry of Education under grant number 2020R1I1A3058605.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Arunprasath, S.; Annamalai, S. Improving patient centric data retrieval and cyber security in healthcare: Privacy preserving solutions for a secure future. *Multimed. Tools Appl.* **2024**, 1–31. [CrossRef]
2. Wang, T.; Wu, Q.; Chen, J.; Chen, F.; Xie, D.; Shen, H. Health data security sharing method based on hybrid blockchain. *Future Gener. Comp. Syst.* **2024**, *153*, 251–261. [CrossRef]
3. Zhang, J.; Yang, Y.; Liu, X.; Ma, J. An efficient blockchain-based hierarchical data sharing for Healthcare Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7139–7150. [CrossRef]
4. Khan, M.A.; Alhakami, H.; Alhakami, W.; Shvetsov, A.V.; Ullah, I. A smart card-based two-factor mutual authentication scheme for efficient deployment of an IoT-based telecare medical information system. *Sensors* **2023**, *23*, 5419. [CrossRef]
5. Lee, J.; Oh, J.; Kwon, D.; Kim, M.; Kim, K.; Park, Y. Blockchain-enabled key aggregate searchable encryption scheme for personal health record sharing with multi-delegation. *IEEE Internet Things J.* **2024**, *11*, 17482–17494. [CrossRef]
6. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of the Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Volume 24, pp. 457–473. [CrossRef]
7. Chu, C.K.; Chow, S.S.; Tzeng, W.G.; Zhou, J.; Deng, R.H. Key-aggregate cryptosystem for scalable data sharing in cloud storage. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 468–477. [CrossRef]
8. Yang, L.; Li, C.; Cheng, Y.; Yu, S.; Ma, J. Achieving privacy-preserving sensitive attributes for large universe based on private set intersection. *Inf. Sci.* **2022**, *582*, 529–546. [CrossRef]
9. Sucasas, V.; Mantas, G.; Papaioannou, M.; Rodriguez, J. Attribute-based pseudonymity for privacy-preserving authentication in cloud services. *IEEE Trans. Cloud Comput.* **2023**, *11*, 168–184. [CrossRef]
10. Wang, H.; Liang, J.; Ding, Y.; Tang, S.; Wang, Y. Ciphertext-policy attribute-based encryption supporting policy-hiding and cloud auditing in smart health. *Comput. Stand. Interfaces* **2023**, *84*, 103696. [CrossRef]
11. Oh, J.; Lee, J.; Kim, M.; Park, Y.; Park, K.; Noh, S. A secure data sharing based on key aggregate searchable encryption in fog-enabled IoT environment. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 4468–4481. [CrossRef]
12. Cremers, C.J. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols: Tool Paper. In Proceedings of the International Conference on Computer Aided Verification, Princeton, NJ, USA, 7–14 July 2008; pp. 414–418. [CrossRef]
13. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
14. MIRACL Cryptographic SDK. Available online: https://github.com/miracl/MIRACL (accessed on 2 April 2024).
15. Bao, Y.; Qiu, W.; Cheng, X. Secure and lightweight fine-grained searchable data sharing for IoT-oriented and cloud-assisted smart healthcare system. *IEEE Internet Things J.* **2022**, *9*, 2513–2526. [CrossRef]
16. Mamta; Gupta, B.B.; Lytras, M.D. Fog-enabled secure and efficient fine-grained searchable data sharing and management scheme for IoT-based healthcare systems. In *IEEE Transactions on Engineering Management*; IEEE: New York, NY, USA, 2022; pp. 1–13. [CrossRef]
17. Wang, Y.; Zhang, A.; Zhang, P.; Qu, Y.; Yu, S. Security-aware and privacy-preserving personal health record sharing using consortium blockchain. *IEEE Internet Things J.* **2022**, *9*, 12014–12028. [CrossRef]
18. Oh, J.; Lee, J.; Kim, M.; Park, Y.; Park, K.; Noh, S. A secure personal health record sharing system with key aggregate dynamic searchable encryption. *Electronics* **2022**, *11*, 3199. [CrossRef]
19. Trivedi, H.S.; Patel, S.J. Key-aggregate searchable encryption with multi-user authorization and keyword untraceability for distributed IoT healthcare systems. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4734. [CrossRef]

20. Xu, G.; Qi, C.; Dong, W.; Gong, L.; Liu, S.; Chen, S.; Liu, J.; Zheng, X. A privacy-preserving medical data sharing scheme based on blockchain. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 698–709. [CrossRef] [PubMed]
21. Zhang, C.; Luo, X.; Fan, Q.; Wu, T.; Zhu, L. Enabling privacy-preserving multi-server collaborative search in smart healthcare. *Future Gener. Comp. Syst.* **2023**, *143*, 265–276. [CrossRef]
22. Zhang, Y.; Guo, F.; Susilo, W.; Yang, G. Balancing privacy and flexibility of cloud-based personal health records sharing system. *IEEE Trans. Cloud Comput.* **2023**, *11*, 2420–2430. [CrossRef]
23. Peng, G.; Zhang, A.; Lin, X. Patient-centric fine-grained access control for electronic medical record sharing with security via dual-blockchain. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 2908–3921. [CrossRef]
24. Zhang, K.; Zhang, Y.; Li, Y.; Liu, X.; Lu, L. A blockchain-based anonymous attribute-based searchable encryption scheme for data sharing. *IEEE Internet Things J.* **2024**, *11*, 1685–1697. [CrossRef]
25. Jastaniah, K.; Zhang, N.; Mustafa, M.A. Efficient user-centric privacy-friendly and flexible wearable data aggregation and sharing. In *IEEE Transactions on Cloud Computing*; IEEE: New York, NY, USA, 2024. [CrossRef]
26. Yin, H.; Zhao, Y.; Zhang, L.; Qiao, B.; Chen, W.; Wang, H. Attribute-based searchable encryption with decentralized key management for healthcare data sharing. *J. Syst. Architect.* **2024**, *148*, 103081. [CrossRef]
27. Lai, C.; Zhang, H.; Lu, R.; Zheng, D. Privacy-preserving medical data sharing scheme based on two-party cloud-assisted PSI. *IEEE Internet Things J.* **2024**, *11*, 15855–15868. [CrossRef]
28. Lax, G.; Nardone, R.; Russo, A. Enabling secure health information sharing among healthcare organizations by public blockchain. *Multimed. Tools Appl.* **2024**, 1–17. [CrossRef]
29. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
30. Patranabis, S.; Shrivastava, Y.; Mukhopadhyay, D. Dynamic key-aggregate cryptosystem on elliptic curves for online data sharing. In *Progress in Cryptology, Proceedings of the INDOCRYPT 2015: 16th International Conference on Cryptology in India, Bangalore, India, 6–9 December 2015*; Springer: Berlin/Heidelberg, Germany, 2015. [CrossRef]
31. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory (TOCT)* **2014**, *6*, 13. [CrossRef]
32. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
33. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of blockchain-based lightweight V2I handover authentication protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
34. Attir, A.; Naït-Abdesselam, F.; Faraoun, K.M. Lightweight anonymous and mutual authentication scheme for wireless body area networks. *Comput. Netw.* **2023**, *224*, 109625. [CrossRef]