eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

E-Tenon: An Efficient Privacy-Preserving Secure Open Data Sharing Scheme for EHR System

Prosanta Gope [a], Zhihui Lin [a,*], Yang Yang [b] and Jianting Ning [c]

[a] *Department of Computer Science, University of Sheffield, S10 2TN Sheffield, U.K.*
E-mails: zhihuilin111@gmail.com, p.gope@sheffield.ac.uk
[b] *Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583, Singapore*
E-mail: y.yang@u.nus.edu
[c] *College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China*
E-mail: jtning88@gmail.com

Abstract. The transition from paper-based information to Electronic-Health-Records (EHRs) has driven various advancements in the modern healthcare industry. In many cases, patients need to share their EHR with healthcare professionals. Given the sensitive and security-critical nature of EHRs, it is essential to consider the security and privacy issues of storing and sharing EHR. However, existing security solutions excessively encrypt the whole database, thus requiring the entire database to be decrypted for each access request, which is time-consuming. On the other hand, the use of EHR for medical research (e.g., development of precision medicine and diagnostics techniques) and optimisation of practices in healthcare organisations require the EHR to be analysed. To achieve that, they should be easily accessible without compromising the patient's privacy. In this paper, we propose an efficient technique called E-Tenon that not only securely keeps all EHR publicly accessible but also provides the desired security features. To the best of our knowledge, this is the *first work* in which an Open Database is used for protecting EHR. The proposed E-Tenon empowers patients to securely share their EHR under their own multi-level, fine-grained access policies. Analyses show that our system outperforms existing solutions in terms of computational complexity.

Keywords: Open database, E-tenon, ABE, Multi-level attribute-based encryption, Multi-signature

1. Introduction

With the rapid development of Health Information Technology (HIT) and cloud services, many healthcare organisations are accelerating the implementation of Electronic Health Record (EHR) based systems. These systems enhance their services and core competencies since EHRs can address many limitations of traditional paper-based medical records, such as scalability, accessibility, and persistence. EHRs are often shared across doctors and healthcare providers with patients' consent. They typically include sensitive and private information such as patient's identity codes, health history, medical diagnoses and treatment plans. Leakage of these data can cause embarrassment or even result in life-threatening consequences for patients. Indeed, despite record levels of security spending by different

---

*Corresponding author. E-mails: zhihuilin111@gmail.com, p.gope@sheffield.ac.uk.

hospitals, there is still a wide range of malicious cyberattacks intended to penetrate databases and connected systems. This is because cybercriminals find EHRs highly profitable, which motivates them to steal such data. Therefore, designing a system that preserves patient privacy robustly and efficiently is imperative.

Motivation. For the application scenarios mentioned above, many existing schemes are vulnerable and ineffective. For example, a common approach recommended and practised in the industry by many security practitioners is to encrypt databases strictly so that the data is protected to the maximum extent possible, even in the event of a security incident. However, it is worth noting that recent trends and incidents such as the COVID-19 outbreak has caused a sharp increase in the volume of medical information held by hospitals. As a result, it is inefficient to excessively encrypt the whole database or a majority of the data since it will have a marked impact on the performance of the EHR system. Besides, EHRs are increasingly being used for developing customised and precision medicine regimens, inventing new and more accurate techniques for diagnosis and treatment, and optimising medical processes to help healthcare organisations to meet growing medical demands, improve operations, and reduce costs. Such applications require the EHRs to be easily accessible for analysis without compromising privacy.

A naive solution to the above requirements is to use Attribute-Based Encryption (ABE), which provides confidentiality and fine-grained access control. There are two general types of ABE: Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [1] and Key-Policy Attribute-Based Encryption (KP-ABE) [2]. In CP-ABE, data is encrypted with a user-defined access structure, and a user with the relevant attributes can decrypt it [3]. Contrarily to CP-ABE, KP-ABE encrypts data with a set of descriptive attributes, and a user with a key embedded with an appropriate access structure can decrypt the data [3]. In this paper, we focus on CP-ABE. As an example, suppose a data owner (patient) wants to share *part* of their EHR with a specific healthcare professional with a specific role and responsibility according to a CP-ABE access policy defined by the patient. One doctor is assigned a specific set of attributes (e.g., { *"doctor", "temporary", "oncology", "top secret"*}) while another doctor may be assigned different attributes (e.g., { *"doctor", "in charge", "dentistry", "confidential"*}).

A doctor is authorised to access a patient's EHR data if his/her attribute set satisfies the access policy for that *part* of the data. Readers may have noticed that naive CP-ABE does not perfectly support multi-level access control, meaning that data owners have to individually set different access policies for different parts of EHRs, depending on the type and sensitivity. As a result, these access policies will introduce many *duplicate attributes*. The number of access policies is proportional to the number of duplicate attributes. That is, as more access policies are registered in the system, there will be more duplicate attributes, which will undoubtedly increase the storage and communication overheads. Worse still, given that most EHR data is sensitive, encrypting and decrypting large volumes of sensitive EHR using naive CP-ABE can be prohibitively expensive (i.e., it suffers from *linear decryption cost* [4]), especially for resource-constrained devices. This, however, yields the following questions: (1) Can we retain the benefits of CP-ABE for fine-grained access control while avoiding duplication of attributes when implementing multi-level access control in EHR systems? (2) Can we protect the confidentiality of EHR data without relying on extensive encryption (i.e., *securely keep most of the EHR data open/in plaintext*)?

Another concern is related to the integrity and authenticity of EHRs. Apart from the information provided by the medical staff, nowadays, more and more health data are collected from connected sensors, and wearable medical devices [5] over (insecure) networks. All of this is patient-centred data, over which *the patient has primary control*. However: (1) what if the patient takes advantage of their primary control to share/upload false data to the EHR system? (2) What if a medical device exploits its
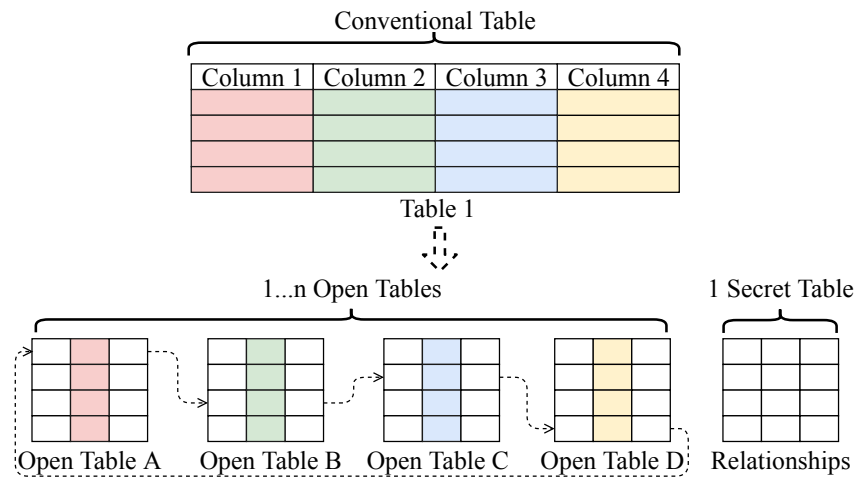
Fig. 1. Overview of the proposed tenon database. A conventional table will be segmented into a series of sub-tables where the relationship between rows is hidden. It can be revealed partially or fully, depending on the data user's attributes (access rights).

automated nature to upload false data to the EHR system without the patient's consent? (3) What if a man-in-the-middle intercepts and tampers with the data?

1.1. Our Contributions

Indeed, strict security requirements appear to be diametrical to the goal of keeping data open. This paper makes a novel attempt to address these seemingly contradicting requirements. It proposes a novel E-Tenon system where data are stored in an open database while maintaining all privacy and security properties.

One of the core components of the proposed system is the Tenon database (TDB), whose overview is presented in Fig. 1. Unlike conventional databases, the TDB is an open database consisting of a series of public tables and one secret table. Its main advantage lies in the fact that data protection does not depend on heavy encryption and decryption. Instead, the protection of EHRs is achieved through data preprocessing, maintenance of secret relationships between EHR blocks, and shuffling techniques. Notably, EHRs will be classified into identifiable information and Non Personally Identifiable Information (Non-PII), the latter of which will be tokenised into EHR blocks and can be securely made public. In addition, EHRs in the TDB are constantly shuffled, which makes it extremely difficult for attackers to exploit the open data. The main contributions of this paper are summarised as follows:

- We design an efficient open database where the majority of the data is open, minimising encryption operations.
- We propose a novel mechanism that does not rely on any suppression and generalisation techniques used in the existing schemes (such as k-Anonymity), where a suppression may lead to data loss and reduced usability, while generalisation may overlook some details about the data.
- We present data preprocessing and shuffling methods used in conjunction with the proposed E-Tenon system to store and share EHRs securely in an open database setting.
- We show how to ensure that a medical device and a data owner sign the same content, even if the EHRs have been preprocessed. This guarantees the authenticity and integrity of EHRs.

Our work addresses the shortcomings of previous solutions since E-Tenon not only efficiently guarantees multi-level, fine-grained EHR-data sharing but also protects the integrity and authenticity of the EHR, most importantly, *under an open database setting*. It takes only 2.34 milliseconds for signing and verifying the signature, and 0.14 and 0.76 seconds for encryption and decryption of the secret relationship, respectively. To the best of our knowledge, E-Tenon is the first open database-based scheme to provide such a wide range of security and privacy properties. Note that while this work focuses on EHR, the concept of E-Tenon would also be applicable in other scenarios requiring low-latency access to user data, such as in mobile edge computing environments.

1.2. Related Work

Since medical data security has become a growing public concern, a considerable number of schemes have been published for secure medical data sharing and privacy preservation [4, 6–17]. For instance, most research in protecting medical data have emphasised the use of cryptographic methods such as CP-ABE and KP-ABE [9, 10, 13, 14, 16, 18]. The system architecture proposed in [6] is based on a successor of CP-ABE and Role-Based Access Control (RBAC) to protect EHR stored in the hybrid cloud with direct and indirect access. In Li et al.'s KP-ABE-based model [12], the data owner needs to trust the key issuer because they are only inserting a set of descriptive attributes into the data using KP-ABE, but they do not know who will be accessing their data [1]. Xu et al. [9] presented a practical dual-policy ABE scheme for EHR systems that combines the advantages of CP-ABE and KP-ABE with support for user revocation. In addition, Belguith et al. [14] proposed a multi-authority CP-ABE scheme that delegates expensive computing tasks to cloud servers, and their scheme also prevents collusion between the authorities.

Nevertheless, no existing solution in the literature is designed for private database settings that can ensure EHR security and patient privacy while keeping data in plaintext form. When every second matters during an emergency, the time-consuming encryption and decryption operations in a healthcare information system may cause delays in accessing patient information (such as medical history) during the golden hour that saves a patient's life. Likewise, as argued in [19], excessive security may obstruct sensible data use by healthcare providers and patients. Most approaches have failed to properly weigh the patients' right to privacy against the legitimate sharing of data. Alternatively, we also analysed the feasibility of applying anonymisation techniques such as t-closeness and Attribute-Based Credentials (ABCs) [20–22] under open database environments. T-closeness is a privacy-preserving technique used for anonymising datasets to prevent the identification of specific individuals based on certain attributes (such as age, income, and medical conditions). It prevents individuals from being identified on the basis of their sensitive attributes by ensuring that the distribution of a sensitive attribute in a group of records is similar to its distribution in the population as a whole. On the other hand, ABCs are a digital identity verification technique that allows users to verify their identities to third parties without revealing too much personal identity information. However, our proposed scheme aims to achieve efficient EHR sharing in a multi-party healthcare setting, and these two techniques are not directly applicable to our proposed solution. Specifically, t-closeness provides a means of anonymising data to prevent the disclosure of sensitive information. In contrast, ABCs provides user authentication, but neither technique provides fine-grained access control or multi-party data sharing, which are key features of our proposed solution. Moreover, although existing anonymisation techniques such as k-anonymity [23] and l-diversity [24] have been extensively studied theoretically and empirically, these widely-adopted principles are still insufficient to prevent attribute disclosure if the attacker has partial knowledge

about the overall sensitive data distribution. On the other hand, although the t-closeness principle has been proposed to address this problem, it can only support sensitive numerical attributes. In addition, most state-of-the-art suppression or generalisation-based anonymisation techniques intentionally remove some parts of the attributes from the database in order to make a particular attribute private, but this is prohibited in our system as we do not remove or modify any patient data. Considering the goals and characteristics of our proposed scheme, we believe these anonymisation solutions are not the optimal choices when compared with ABE techniques.

Although several similar works mentioned above have used ABE to protect EHR, which is promising for flexible and fine-grained EHR sharing, they are computationally intensive when applied to encrypt the entire database. In addition, most solutions cannot support searching over encrypted data directly. Consequently, to search for relevant patient data in an encrypted database, the system first needs to decrypt the data on the application back-end. Such a burdensome process wastes valuable computing resources. Furthermore, many schemes fail to use digital signatures to ensure data integrity and authenticity properly. For example, [16] allows only one entity to sign the EHR, which grants the entity too much power. Despite some schemes [25] allowing multiple entities to sign the data, they cannot guarantee that the same content is being signed honestly by all participants.

To our knowledge, no state-of-the-art work on sharing and protecting EHRs has considered using a secure open database to save the avoidable overhead of encryption and decryption. That said, as the current solutions are built on private databases by default, we are unable to find related work that fully meets our expectations.

1.3. Organisation

The rest of the paper is organised as follows. Section 2 introduces and recapitulates the required mathematical notations, security assumptions and related schemes. Section 3 presents the system model and the corresponding adversarial model. This is followed by the construction of E-Tenon, given in detail in Section 4. Next, we prove the security and practicality of the proposed scheme by conducting security and performance analysis in Sections 5 and 6, respectively. Section 7 of the paper concludes our work in light of all that has been mentioned.

2. Preliminaries

This section introduces and recapitulates several prerequisites, including definitions of some mathematical notations, a multi-level ABE scheme, and a multi-signature scheme.

2.1. Notations

We use $r \xleftarrow{\$} \mathbb{R}$ to mean that r is chosen at random from $\mathbb{R}$, and $o \leftarrow A(i_1, i_2, \cdots, i_n)$ to denote an algorithm A that takes i_1 to i_n as input parameters and yields the outcome of its operation o. If an algorithm returns $\perp$, it symbolises that the algorithm has failed to perform the expected actions ($v(\perp) = False$). $\mathbb{Z}_p$ is the set of integers modulo p, such that $\mathbb{Z}_p = \{[0]_p, [1]_p, \cdots, [p-1]_p\}$. $\mathbb{G}$ is a multiplicative group of prime order p where $0 \notin \mathbb{G}$ since the multiplicative inverse of 0 does not exist. In addition, we denote $\mathbb{G} \backslash \{1\}$ by $\mathbb{G}^*$.

2.2. Building Blocks

More formal definitions are provided below. Bilinear maps are a helpful tool for pairing-based cryptography because they conveniently establish relationships between cryptographic groups. As cyclic groups are used in the bilinear map, we first introduce the definition of a cyclic group.

Definition 1 (Cyclic Group of Prime Order [26, 27]). *Let $\mathbb{G}_0 = \langle g \rangle$ be a cyclic group of prime order p where $\langle g \rangle = \{g^n : n \in \mathbb{Z}\}$, generator $g \in \mathbb{G}_0$, and p is a k-bit integer. Note that $\mathbb{G}_0$ can be denoted multiplicatively, and $\langle g \rangle$ is a cyclic subgroup of $\mathbb{G}_0$ generated by g.*

Definition 2 (Prime Order Bilinear Group [1]). *Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of the same prime order p. g is an arbitrary generator $g \xleftarrow{\$} \mathbb{G}_0$. e is a symmetric bilinear map, such that $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$ where $e(g^x, g^y) = e(g^y, g^x) = e(g,g)^{xy} = e(g,g)^{yx}$.*

There are three properties of an efficiently-computable *e* that are worth noting:

(1) *Bilinearity*: $e(g^y, g^x)$ and $e(g,h)^{xy}$ must be equivalent for all $x, y \xleftarrow{\$} \mathbb{Z}_p$ and $g_i, g_j \xleftarrow{\$} \mathbb{G}_0$.
(2) *Non-degeneracy*: $e(g,g)$ must not be equal to the identity of $\mathbb{G}_1$.
(3) *Computability*: for all $g, h \xleftarrow{\$} \mathbb{G}_0$, there exists an algorithm that can efficiently compute $e(g,h)$.

Definition 3 (Discrete Logarithm Assumption [27]). *Let $\mathbb{G}_0$ be a multiplicative cyclic group with a prime order p and a generator g. The advantage is formulated as follows when a Probabilistic Polynomial-Time algorithm $\mathcal{A}$ is applied to solve the discrete logarithmic problem in $\mathbb{G}_0$:*

$$\text{Adv}_{\mathbb{G}_0}^{dlog}(A) = \Pr\left[g^x = y | g \xleftarrow{\$} \mathbb{G}_0^*; y \xleftarrow{\$} \mathbb{G}_0; x \xleftarrow{\$} A(y)\right]$$

The assumption holds when $\text{Adv}_{\mathbb{G}_0}^{dlog}(\mathcal{A})$ is negligible.

2.3. Multi-level CP-ABE

ABE is an outstanding example of flexible and scalable encryption mechanisms for multiple users in recent years. It enables contextualised decision-making thanks to the introduction of the concept of attributes. In KP-ABE [2, 3], data owners have a set of attributes closely linked to themselves that can be selectively applied to encrypt their data. Any other user who intends to decrypt the ciphertext first needs to be issued a key bundled with a suitable access structure by the trusted key issuer. In contrast, the data owner gains more control in CP-ABE regarding who can access his/her data, through the design of an access policy embedded in the ciphertext [1]. The only users who can access and decrypt this data are those with the appropriate attributes. Therefore, CP-ABE is probably better suited for data outsourcing, especially when it is used to preserve patients' privacy, even in emergencies, due to its flexible nature. However, in the challenging field of E-health, the standard CP-ABE is not perfectly compatible with the reality of the intertwined doctor-patient relationships among different healthcare organisations. This is because each distinct part of the EHR file may require to be accessed with completely different access rights depending on the purpose of the data user. Therefore, the naive CP-ABE is not fully compatible in our scenario. Fortunately, as one of the successors to CP-ABE, ML-ABE fills in the gaps and imperfections described above.

Definition 4 (ML-ABE [28])**.** *ML-ABE consists of four algorithms (setup, encrypt, keygen, decrypt):*

- ***setup****: This algorithm is executed by a trusted authority to generate public parameters $\mathfrak{pp}$ and a master key $\mathfrak{msk}$ according to the security parameter $\mathfrak{K}$.*
- ***encrypt****: It is invoked by the data owner to encrypt the plaintext $\mathbb{M} = \{m_l\}_{l \in \{1,c\}}$ concerning the multi-level security, where c represents the number of security levels. There are four required inputs, $\mathfrak{pp}$, the plaintext $\mathbb{M}$, the access tree $\mathbb{A}$ defined by the data owner over the universe of attributes $\mathbb{S}$, and the set of security levels $\{k_l\}_{l \in \{1,c\}}$. It returns the enciphered data $\mathbb{C} := \{\mathbb{A}, \forall k_l : \{\mathbb{A}''_i\}_l, \mathbb{C}_l\}$. Here we underline that $\{\mathbb{A}''_i\}_l$ is a set of required sub-trees that must be satisfied by each security level k_l for $l \in \{1,c\}$. We also define their access structure below.*
- ***keygen****: This algorithm is performed by the trusted authority to generate and issue the decryption key for the users depending on a set of attributes $\mathbb{S}$. It takes as input a set of attributes $\mathbb{S}$, the public parameters $\mathfrak{pp}$, and the master key $\mathfrak{msk}$ generated previously. The output will be the corresponding decryption key $\mathcal{DK}$ for a specific user or entity involved in the system.*
- ***decrypt****: This algorithm is called by the data user to decrypt the ciphertext with respect to the multi-level security. There are three required inputs, the public parameters $\mathfrak{pp}$, the ciphertext $\mathbb{C}$, and the decryption key $\mathcal{DK}$. Here $\mathbb{C}$ is packed with the relevant access policy $\mathbb{A}$, the security level k_l, and a set of required sub-trees $\{\mathbb{A}''_i\}_l$. It outputs the plaintext m_l by decrypting the corresponding ciphertext $\mathbb{C}_l$ if the deciphering entity's attributes meet the requisites described in $\mathbb{C}$.*

Definition 5 (Access Structure [28])**.** *Let $\mathbb{A}$ be the access structure with multi-threshold security levels k_l, $l \in \{1,c\}$. Let $\mathbb{A}'_x$ be the sub-tree of $\mathbb{A}$ rooted at a particular node x. Also, let $\{\{\mathbb{A}''_i\}_l\}$ be the sub-trees within the outer level. The root node is an AND gate defined as a k_l-out-of-c security levels. p_l subsets of attributes and n_l sub-trees of the root node are required to reconstruct the corresponding secret sharing embedded in the ciphertext $\mathbb{C}$ for security level k_l. $\mathbb{A}'_x(\mathbb{S}) = 1$ if and only if a set of attributes $\mathbb{S} = \{a_i\}_{i \in \{1,l\}}$ satisfies the sub-tree and the number of attributes l is at least as many as the number of children of node x, otherwise $\mathbb{A}'_x(\mathbb{S}) = \bot$.*

2.4. Multi-Signature

A Multi-Signature (MS) solution allows a group of signers to co-sign on a shared document in a compact manner [27]. To provide a real-life example, publishing a report/document often requires the cooperation of multiple colleagues. In order to guarantee the authenticity of the report, each participant must sign the file. Therefore, Multi-Signature technology is used to fulfil this type of requirement in the electronic world. Besides, the ABE approach described in the previous section has already reduced the cost of key management by providing one-to-many encrypted access control [29]. Thus, we prefer to use a Multi-Signature scheme that is not based on comparatively more burdensome requirements of PKI (e.g., knowledge of secret key hypothesis [30]) to enhance the practicality of the proposed E-Tenon system further. Bellare and Neven's MS-BN [27] defined below fits well with our concept.

Definition 6 (MS-BN [27])**.** *MS-BN is a scheme consisting of four randomised algorithms (Pg, Kg, Sign, Vf):*

- ***Pg****: This algorithm is executed by a trusted authority to generate global parameters and output $\mathbb{G}, p, g$, where $\mathbb{G}$ is a multiplicative cyclic group of prime order p, and g is a generator of $\mathbb{G}$ chosen at random.*

- **Kg**: *This algorithm is called by each signer and co-signer to produce their own key pair used in the signing process. It outputs the signing key $\mathcal{SK} := r \overset{\$}{\leftarrow} \mathbb{Z}_p$ randomly chosen from the finite field $\mathbb{Z}_p$ and the related verification key $\mathcal{VK} := g^{\mathcal{SK}}$.*
- **Sign**: *This algorithm is performed by the signers, and there are three rounds of communication. Each signer will perform some computation in the local scope based on messages shared by all co-signers as well as share their own message with others. It takes as input a signing key of the current signer $\mathcal{SK}_i$, a list of verification keys of all involved signers $\mathbb{V} := \{\mathcal{VK}_1, \mathcal{VK}_2, \cdots, \mathcal{VK}_n\}$, and a message $\mathfrak{msg}$ to be multi-signed. It outputs the compact signature σ consisting of the nonce commitments and the signatures if everyone is honest; otherwise, it outputs $\perp$.*
- **Vf**: *This algorithm is executed by the verifiers. There are three required inputs: a message $\mathfrak{msg}$, a compact signature to be verified σ, and a set of verification keys of all involved signers $\mathbb{V} := \{\mathcal{VK}_1, \mathcal{VK}_2, \cdots, \mathcal{VK}_n\}$. It returns 1 to indicate the signature σ is valid; otherwise, it returns $\perp$.*

Here we stress two essential facts about MS-BN. First, the security of this scheme is guaranteed on the assumption that at least one of the signers is honest [27, Sec. 4]. Second, the *Kg* algorithm of MS-BN is run independently by each signer to generate the key pair. Such an assumption leads to a security breach when all the signers are honest-but-curious or dishonest. Given the increasing sophistication of cyber attacks, any end-user can no longer be undoubtedly trusted. Hence, our model will strengthen MS-BN to accommodate cases where no particular signer is fully trusted. To achieve that, we do not allow the non-trusted signer to perform the *Kg* algorithm without the support of a trusted entity. In other words, the secret keys required for the user to operate the ABE and Multi-Signature related algorithms will be issued by an Attribute Authority (AA) at once where necessary.

2.5. E-Tenon

We propose the Electronic Tenon System (E-Tenon), depicted in Fig. 2, which effectively integrates Multi-level CP-ABE and Multi-Signature techniques. Our novel innovations and extensions enable these existing technologies to function optimally within open database environments. To our knowledge, current ABE-based privacy-preserving systems incur significant encryption and decryption overheads. However, our solution confidently allows EHRs to be securely opened in the database after special preprocessing. Specifically, EHR blocks stored within the database can only be mapped into meaningful information by deciphering relevant secret pointers. Data shuffling techniques are employed to constantly change the position and order of EHR blocks, ensuring that open data is randomly presented to data users each time the database is accessed. Furthermore, in the original MS-BN scheme, there must be a trusted signing entity involved in the signing process, but we cannot assume that this will be feasible in safety-critical applications. Therefore, our solution does not require a fully trusted signer to ensure multi-signature unforgeability, making our E-Tenon system more flexible and practical. Eventually, we present steps grounded on sound logic in this paper to guarantee that the service providers and data owners can consistently sign the same message. These features empower us to manage EHRs efficiently, flexibly and granularly while preserving privacy and security.

3. System and Adversarial Model

In this section, we provide a high-level overview of the proposed system model with respect to entities involved in E-Tenon. Afterwards, we analyse security considerations along with an adversarial model.
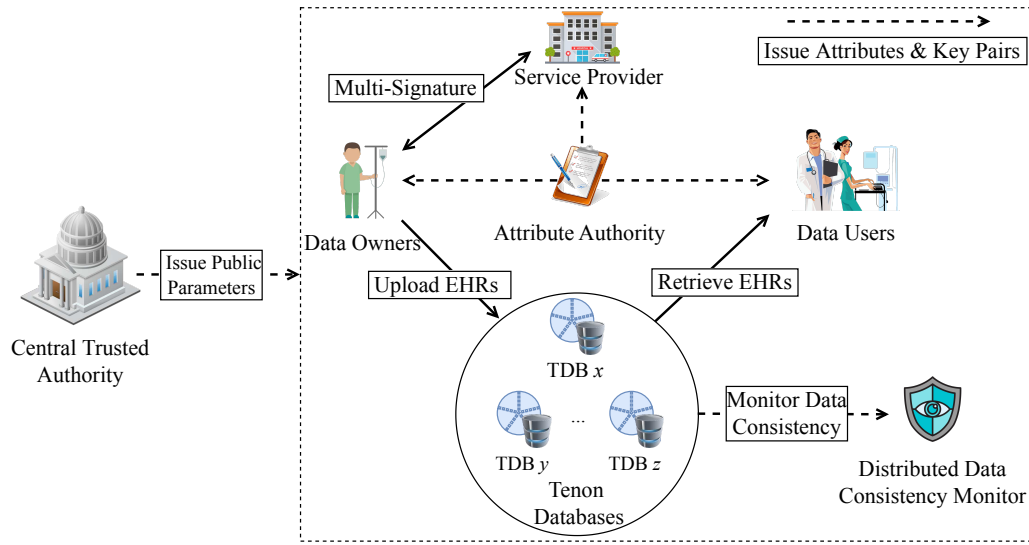
Fig. 2. System model of the proposed scheme.

3.1. System Model

To establish the system model, we first introduce an efficient open database, then we merge and extend a Multi-Signature scheme MS-BN [27] with an encryption scheme ML-ABE [28]. Our system (as depicted in Fig. 2) ends up with three distinct phases: SETUP, ACCUMULATION and RETRIEVAL, along with seven secure algorithms. In addition, there are six crucial entities: Central Trusted Authority (CTA), Attribute Authority (AA), Data Owner (DO), Service Provider (SP), Data User (DU), and Tenon Database (TDB). Besides, we allow for the option of a seventh participant: Distributed Data Consistency Monitor (DDCM).

- **CTA** is a fully trusted entity responsible for generating system-wide public parameters for all participants within the system. In practice, it is typically operated by national governing bodies or governmental organisations, such as the National Health Service (NHS) in the UK.
- **AA** is managed by the same administrative team as the CTA. It is in charge of maintaining the user's attributes and issuing secret keys for the user, where appropriate. We note that the state-of-the-art multi-authority ABE systems use several different AAs, making each AA responsible for only one specific attribute. However, it must resist collusion attacks. In our case, we do not require multiple AAs and consider the AA a trusted entity.
- **DO** is the actual owner of the EHRs, i.e., the patient. Typically, DOs are concerned about the privacy of their EHRs, and they have the right to control the sharing of their EHRs. However, DOs can also be malicious. For example, DOs may upload incorrect EHRs to mislead data users into making improper treatment decisions. In E-Tenon, DOs can preprocess and selectively encrypt EHRs with self-defined multi-level access policies before sending the data to the database. DOs will also be required to multi-sign their data.
- **SP** is an entity (such as a doctor and a hospital/clinic) responsible for providing patients with diagnostic results, as well as collecting the reading (or measurement) from smart IoT/monitoring devices, such as smart blood pressure sensors, and uploading them to the database with the DO's

consent. This type of entity is generally assumed to be trusted, but in some special cases, it may be considered malicious if it manipulates the data before uploading it. Please note that doctors can serve dual roles as DU, who access their patient's medical history and SP, who upload diagnostic reports after consultations or treatments are completed.

- **TDB** is an honest-but-curious entity responsible for the data management. TDB is a distributed open database. The data should be stored as it is, and TDB has no right to decrypt any secret relationships. We are inspired by the ancient timber mortise and tenon joints, a strong and stable way of joining multiple elements together by using a proper combination of concave and convex pieces, as shown in Fig. 3, when designing the TDB and introducing the *Electronic Tenon Structure* for different EHR blocks to be securely joined together. By secure, we mean that no public data can be exploited by unauthorised entities, as only data users with the appropriate attributes know the proper way to assemble the relevant EHR blocks.
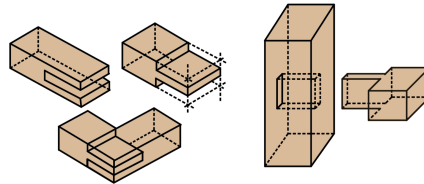

Fig. 3. An example of mortise and tenon joints.

- **DU** is an individual or organisation (e.g., doctor, hospital, research institution, pharmaceutical and medical insurance company) that needs access to patient-owned EHRs in the TDB. DU requires an appropriate level of access, represented by their attributes, to reveal the secret relationships between EHR blocks. For example, a doctor may be able to extract five secret pointers to find and link five EHR blocks. However, a nurse may only be able to decrypt two pointers. Thus, there is a restriction on the amount of data that can be recovered due to their different attributes. Moreover, a DU without the required attributes will be considered malicious when attempting to decrypt the pointers.
- **DDCM** is a trusted optional participant responsible for auditing data consistency between multiple TDBs. It is usually a built-in process/algorithm running on the database system that adheres to a predefined protocol incapable of learning anything from encrypted content (e.g., the secret relationships between EHR blocks). The synchronisation of EHR across multiple databases enhances availability and avoids single points of failure.

3.2. Adversarial Model

E-Tenon is intended to be used by patients and a wide range of healthcare institutions. The novelty lies in the fact that most of the EHRs in the TDB are publicly accessible. Besides, we do not restrict EHRs to be transferred only within private networks such as the corporate Local Area Network. Accordingly, the vast majority of EHRs can be transmitted through untrusted public networks such as the Internet. While these considerations significantly increase the applicability and efficiency of the model, they also expose system interactions and EHRs in transit to various malicious cyber attackers. Therefore, our system must defend against the following threats:

- **Confidentiality Threat**: The system may fail to guarantee the secrecy of secret relationships between EHR blocks. For instance, a semi-trusted TDB may intend to discover as much information as possible while complying with the defined protocols. A malicious DU without appropriate permissions may attempt to exploit the open data and reveal secret relationships.
- **Privacy Threat**: DO and DU's identity may be revealed when interacting with a semi-trusted TDB. A malicious DU may infer a relationship between the patient and the data stored in the TDB.
- **Integrity and Authenticity Threat**: As EHR is patient-centric data, the patient has primary control over it. However, ensuring the integrity and authenticity of the EHR provided by patients remains a challenge. One possible attack is that the EHR is tampered with by an intermediary when transmitted over insecure public channels. Even worse, patients themselves may deliberately alter their EHR before uploading in order to obtain *biased* diagnosis and then obtain a large insurance claim (they may also deny that they have uploaded fake data). In this context, although we can use digital signatures to resist these attacks, they may be forged.

3.3. Security Assumptions

Some of the key assumptions are summarised as follows:

- DOs and DUs are expected to be educated about privacy rights and obligations. Thus, they will not actively disclose any confidential information to unaffiliated and unauthorised third parties.
- DOs can apply appropriate access policies to different categories of EHRs according to a *layman-friendly* guidebook provided by the administrator.
- The semi-trusted TDB and unauthorised DUs cannot infer the data type of EHRs when each data category contains at least κ different data types.

3.4. Security Games

Based on the system and adversarial models, we consider the following security games to define the security notion of our E-Tenon system.

1) To prove that E-Tenon is secure against confidentiality and privacy threats, we define an IND-CCA-1 security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

- **Setup:** $\mathcal{C}$ runs setup algorithm, and sends the public parameters $\mathfrak{pp}$ to $\mathcal{A}$.
- **Query:** $\mathcal{C}$ initialises an empty table T, an integer session counter j starting from zero and an empty set $\mathbb{Q}$. $\mathcal{A}$ can repeatedly query the following:
    * **Create:** $\mathcal{C}$ increments j by 1. $\mathcal{C}$ runs setup to obtain $\mathfrak{pp}$ and a master key $\mathfrak{msk}$, then it runs keyGeneration to extract a decryption key $\mathcal{DK}$ on $\mathbb{S}$ and the corresponding security levels k_l. $\mathcal{C}$ finally stores the entry $(j, \mathbb{S}, \mathfrak{pp}, \mathfrak{msk}, \mathcal{DK})$ in T if it is not a duplicate entry.
    * **Corrupt:** $\mathcal{A}$ requests the decryption output of a ciphertext $\mathbb{C}$ using $\mathcal{DK}$ on $\mathbb{S}$. $\mathcal{C}$ sets $\mathbb{Q} = \mathbb{Q} \cup \mathbb{S}$ if the $\mathcal{DK}$ for $\mathbb{S}$ exists in T and proceeds.
    * **Decrypt:** $\mathcal{C}$ decrypts $\mathbb{C}$ and outputs the results of the decryption to $\mathcal{A}$. Note this oracle can only be accessed before $\mathcal{A}$ receives the challenge ciphertext.
- **Challenge:** $\mathcal{A}$ chooses two plaintext message $\mathbb{M}_0$ and $\mathbb{M}_1$ of the same length. $\mathcal{A}$ also submits a challenge access structure $\mathbb{A}^*$ such that $\mathbb{S}$ does not satisfy $\mathbb{A}^*$ for all $\mathbb{S} \in \mathbb{Q}$. $\mathcal{C}$ then randomly selects a bit $b \in \{0, 1\}$ and outputs the encryption results of $\mathbb{M}_b$ under $\mathbb{A}^*$ and k_l to $\mathcal{A}$.

- **Guess:** $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ for b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 7. *ML-ABE is CCA-1 secure against confidentiality and privacy threats, if for all PPT adversaries, there is a negligible function in winning the security game defined above, such that*

$$\mathrm{Adv}_{\mathcal{A}}^{CCA-1}(\lambda) = \Pr[b' = b] = \frac{1}{2} \pm \epsilon$$

2) To prove that E-Tenon is secure against integrity and authenticity threats, we define an MU-UF-CMA security game between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$:

- **Setup:** $\mathcal{C}$ runs setup and keyGeneration algorithms, and sends the public parameters $\mathfrak{pp}$, a random secret key $\mathcal{SK}^*$ and a public key $\mathcal{VK}^*$ to an honest signer. $\mathcal{VK}^*$ is also shared with $\mathcal{F}$.
- **Attack:** $\mathcal{F}$ initialises a message $\mathfrak{msg}$ to be multi-signed and a set containing the public keys of all co-signers $\mathbb{V} = \{\mathcal{VK}_1, \cdots, \mathcal{VK}_n\}$ where $\mathcal{VK}^* \in \mathbb{V}$. Note that all keys in $\mathbb{V}$ are controlled by $\mathcal{F}$ except for $\mathcal{VK}^*$. Meaning that $\mathcal{F}$ impersonates other co-signers with these keys to run the multiSign algorithm with the honest signer. It either outputs a signature σ or a $\perp$.
- **Forgery:** Once the above phase terminates, $\mathcal{F}$ outputs its forgery $(\mathbb{V}, \mathfrak{msg}, \sigma)$. $\mathcal{F}$ wins the game if the forgery passes the verify algorithm.

Definition 8. *MS-BN is MU-UF-CMA secure against integrity and authenticity threats, if for all PPT adversaries, there is a negligible function in winning the security game defined above, such that*

$$\mathrm{Adv}_{\mathcal{F}}^{MU-UF-CMA}(\lambda) = \Pr[\mathrm{verify}(\mathbb{V}, \mathfrak{msg}, \sigma) = 1] \leqslant \epsilon$$

4. Concrete Construction

Our system incorporates three important phases and seven secure algorithms. We describe the construction details of each phase separately, with further specifications in the following subsections.

4.1. Workflow of E-Tenon

The workflow of our E-Tenon system is presented in Fig. 4 where green entities are fully trusted, red entities can be malicious, and blue entities are honest-but-curious. During the SETUP phase, the CTA and AA will generate and issue the public parameters, attributes and keys required by all system users. In the next stage, named ACCUMULATION, a total of four fundamental algorithms are used. Before the secret relationships between EHR blocks are established, they can be classified into two main categories: identifiable and Non-PII data, based on the attributes (such as social security number, medical record number) listed in Health Insurance Portability and Accountability Act (HIPAA). The EHR preprocessing algorithm will detect any HIPAA attributes; these can be tokenised into smaller chunks on a specific level to render them unidentifiable, or minor encryption can be applied to the detected HIPAA identifiers if requested by the data owner. Then, HIPAA identifiers and Non-PII data will be made open after preprocessing as it can not be used to trace a patient's identity without the ability to read and understand the secret relationships between EHR blocks. Note that when encryption is performed with a patient-defined access policy, it is equivalent to the patient giving consent to those
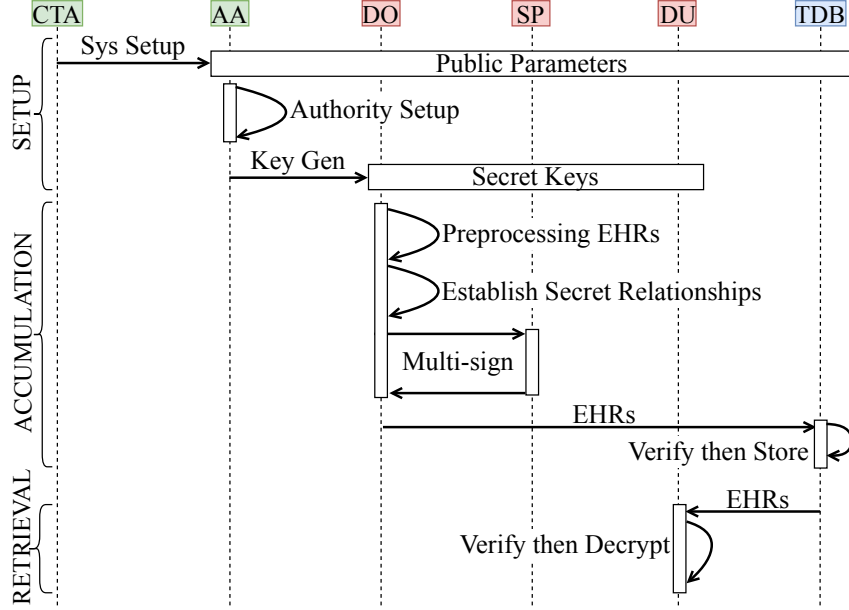
Fig. 4. Workflow of the proposed scheme.

users who satisfy the access policy. Upon multi-signing the data by the DO and SP, the TDB may refuse to store the data if the signature is invalid or forged. Apart from this, signers may also refuse to sign if they believe the data is illegally modified. At the final RETRIEVAL stage, the DUs also have the option to verify the data's signature. They can decrypt the pointers at different security levels according to their attributes when they believe the signature is legitimate. Then, the decrypted pointers can be used to find and combine the relevant EHR blocks in the proper order to recover the correct information.

4.2. SETUP Phase

Table 1 lists some essential notations and cryptographic functions we used. Let λ be the implicit security parameter that denotes the size of the cryptographic groups, and let $\mathbb{S} := \{a_1, a_2, \cdots, a_n\}$ be the universe of the entity's attributes. The following two algorithms need to be administered by the CTA and AA for the initial system and authority setup process of the proposed scheme.

setup(λ): It initially selects a generator $g \xleftarrow{\$} \mathbb{G}_0^*$ and two unique elements γ and δ, at random $\gamma, \delta \xleftarrow{\$} \mathbb{Z}_p$. Then, the master key $\mathfrak{msk}$ is defined as $\mathfrak{msk} := (\delta, g^\gamma)$. Finally, the public parameters $\mathfrak{pp}$ are grouped into the following seven auxiliary elements $\mathfrak{pp} := \{\mathbb{G}_0, \mathbb{G}_1, p, g, g^\delta, e, e(g,g)^\gamma\}$. $\mathfrak{pp}$ then is made public at system level and $\mathfrak{msk}$ can be used to create decryption keys according to user attributes.

keyGeneration($\mathfrak{pp}, \mathfrak{msk}, \mathbb{S}$): This algorithm can be executed by either the AA or the signing parties depending on whether a trusted SP is involved in the signing process or not. In the first case, AA uses this algorithm to produce two *distinct* pairs of keys (i.e., $\mathcal{SDK}$, the signing and decryption key pair, and $\mathcal{VEK}$, the verification and encryption key pair) once $\mathfrak{pp}$ and $\mathfrak{msk}$ are successfully generated by the CTA. It starts by choosing one random $\mathfrak{r}$ and a set of randoms $\{\mathfrak{r}_a\}$ from the finite field $\mathbb{Z}_p$ where each a is in $\mathbb{S}$ such that $\forall a \in \mathbb{S} : \mathfrak{r}, \mathfrak{r}_a \xleftarrow{\$} \mathbb{Z}_p$. These are used to randomise private keys and prevent DOs from compromising data confidentiality by colluding. All necessary keys for the user to operate both ABE

Table 1

Notations and cryptographic functions.

Notation	Definition
$H(\cdot)$	One-way hash function
$\|$	Concatenation operation
$\perp$	Bottom constant of propositional logic
$\{k_l\}_{l \in \{1,c\}}$	Set of security levels
c	Number of security levels
$\mathcal{SK}$	Signing key
$\mathcal{VK}$	Verification key
$\mathcal{EK}$	Encryption key
$\mathcal{DK}$	Decryption key
$\mathcal{SDK}$	Signing and decryption key pair
$\mathcal{VEK}$	Verification and encryption key pair
$\mathbb{A}$	Patient-defined access structure
$\{\mathbb{A}_i''\}_l$	Sub-trees, sub-access structure
$\mathbb{S}$	Universe of attributes
$\mathbb{V}$	Verification key set
$\mathfrak{N}_i$	A unique pointer
Φ_i	A tokenised EHR block
σ	Multi-Signature
$\gamma, \delta, \mathfrak{r}$	Random exponents
ϵ	A negligible number

and Multi-sig algorithms are formed along the following lines:

$$keys := \{\mathcal{SDK} = (\mathcal{SK}, \mathcal{DK}), \mathcal{VEK} = (\mathcal{VK}, \mathcal{EK})\}$$

$$\begin{cases} \mathcal{SK} = \mathfrak{r}, \ \mathcal{VK} = g^{\mathcal{SK}}, \ \mathcal{EK} = \mathfrak{pp} \\ \mathcal{DK} = \left\{ \mathcal{D} = g^{\frac{\gamma + \mathfrak{r}}{\delta}}, \forall a \in \mathbb{S} : \mathcal{D}_a = g^{\mathfrak{r}}.H(a)^{\mathfrak{r}_a}, \mathcal{D}_a' = g^{\mathfrak{r}_a} \right\} \end{cases}$$

where $\mathcal{VK}$ and $\mathcal{EK}$ can be made public, but $\mathcal{SK}$ and $\mathcal{DK}$ need to be kept secret. In the second case, if a trusted SP is involved in the multi-signature process, the signer may choose to generate his/her own signing key pair without relying on the AA. Despite that, $\mathcal{EK}$ and $\mathcal{DK}$ are still required to be issued by an AA, adding an additional layer of security.

4.3. ACCUMULATION Phase

In order to understand what must be encrypted and left open, we need to consider how data may be combined. For instance, an insecure combination is the National Insurance Number (NINO) with the medical condition since it reveals the patient's identity. However, blood pressure and symptoms can be seen as a safe combination. But it is noted that although the knowledge of a single symptom is not helpful in revealing a patient's identity (e.g., almost everyone may have a cough), detailed symptom information can be useful in inferring a patient's identity (e.g., it may be rare for a person to have a nosebleed, cough, fever and heart pain at the same time).

Algorithm 1 dataPreprocessing(Φ, *lv*)

1: **Begin:**
2: **Step 1** classifies Φ as *identifiable* and *Non-PII* in accordance with the HIPAA requirements.
3: **Step 2** If *tokenisable*(identifiers) == true, tokenises identifiers into smaller chunks on a specific level *lv* to render them unidentifiable.
4: **Step 3** If *tokenisable*(Non-PII) == true, splits Non-PII into data blocks, each block contains one main word plus the preceding stopwords if present.
5: **Step 4** establishes the *relationships* between an identifiable column and Non-PII columns as well as the *relationships* between blocks of Non-PII columns.
6: **Step 5** returns preprocessed/structured EHRs $\mathbb{M}$.
7: **End.**

---

dataPreprocessing(Φ, *lv*)): This algorithm is run by the DO. It begins by classifying and labelling EHRs by identifiable and non-personally identifiable information. As an example, identifiable columns include the patient's NINO and mobile number. Non-identifiable columns include medical condition, gender, symptom and blood pressure. Next, it splits any tokenisable identifiers and Non-PII records into chunks/blocks with the relationships between them linked by a 128-bit pointer (UUID). Instead of using pure numeric IDs that are easily guessed, we generate the Universally Unique Identifier using a cryptographically strong pseudo-random number generator provided in the Apache Commons IO library [31]. An example of a 128-bit UUID is *9458fdcc-6bed-46ec-b883-0076409e76f*. This prevents simple brute-force guessing of the secret relationships because it is impossible to iterate through all random UUIDs. In the end, the preprocessed EHR blocks are output in a random access data structure, referred as *electronic tenon structure*: $\mathbb{M} := \{(\mathfrak{N}_1, \Phi_1, \mathfrak{N}_x), (\mathfrak{N}_2, \Phi_2, \mathfrak{N}_y), \cdots, (\mathfrak{N}_n, \Phi_n, \mathfrak{N}_z)\}$. Note that each element in $\mathbb{M}$ is a three-tuple containing (1) the UUID of the current EHR block, (2) the EHR block itself and (3) a pointer to the next EHR block.

To give a more intuitive example, we note that the length of each token *lv* can be customised to the DO's preference, typically consisting of 1-2 characters. An identifiable phone number, *0889623625621*, can be tokenised to [*08,89,62,36,25,62,1*]. In this case, we soon discovered that many duplicate tokens would be stored in the database, taking up a lot of unnecessary storage space. To avoid this, two system tables will be preset in the database. The first system table contains all the numbers from 0 to 9 and all possible two-digit combinations, with 110 numbers. The second system table contains all the letters from a to z and all possible 2-letter combinations from aa to zz, with a total of 702 strings. The entirety of the tokens is located within these two tables, with their corresponding IDs serving to construct a secret relationship that can solely be revealed by a user possessing valid security attributes. In this way, the likelihood of re-identification attacks is negligible, even when identifiable records are openly accessible, as the attacker is unable to decipher the confidential correlation between the numerical and textual EHR blocks.

encryptPointer($\mathfrak{pp}$, $\mathbb{M}$, $\mathbb{A}$, $\{k_l\}_{l \in \{1,c\}}$): This algorithm is executed by the DO. It extracts the pointers $\{\mathfrak{N}_i, \mathfrak{N}_j, \mathfrak{N}_k, \cdots\}$ in $\mathbb{M}$, and encrypts them according to a patient-defined access structure $\mathbb{A}$ with different security levels $\{k_l\}_{l \in \{1,c\}}$, where pointers associated with different security levels require different attributes to decrypt. The ciphertext structure introduced in [28] is adapted as below:

$$\mathbb{C} := \left\{ \mathbb{A}, \forall k_l : \{\mathbb{A}_i''\}_l, \mathbb{C}_{k_l}, \tilde{\mathbb{C}}_{k_l}, \forall y : \mathbb{C}_y, \mathbb{C}_y' \right\}$$
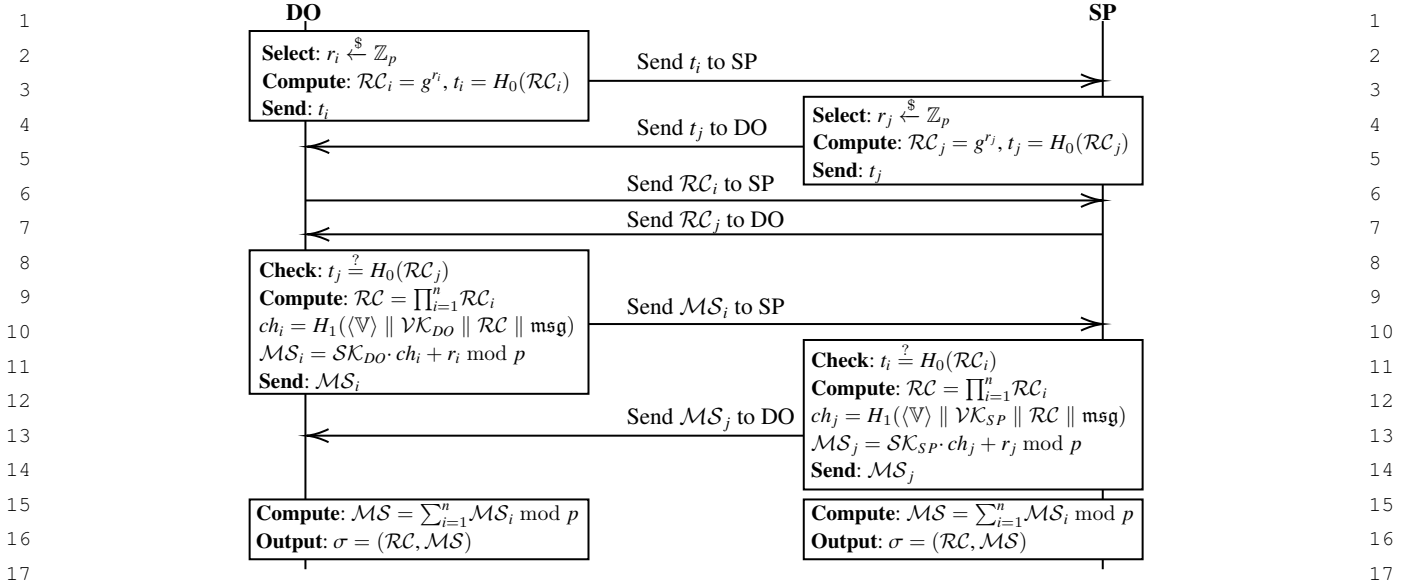
Fig. 5. Rounds of communication in multiSign algorithm (DO stands for Data Owner and SP stands for Service Provider).

$$\begin{cases} \mathbb{C}_{k_l} = g^{\delta \varsigma_l}, \quad \tilde{\mathbb{C}}_{k_l} = \mathfrak{N}_i \cdot e(g,g)^{\gamma \varsigma_l} \\ \mathbb{C}_y = g^{q_y(0)}, \quad \mathbb{C}'_y = H(\text{att}(y))^{q_y(0)} \end{cases}$$

where g^δ and $e(g,g)^\gamma$ are extracted from the public parameters $\mathfrak{pp}$ generated during the SETUP phase by the CTA. Moreover, we note that the advantage of CP-ABE is that the enciphering secret is built into the relevant ciphertext, rather than being placed in the private key (key management is minimised) [1]. Here, the enciphering secret ς_l embedded in each ciphertext with a particular security level k_l is computed as $\varsigma_l := \sum_{i \in \{1,2,\cdots,n_l\}} q_r(\text{index}(x_i))$ where $q_r(x)$ is the polynomial related to the root node r of $\mathbb{A}$, $q_r(x) = a_0 + a_1 x + \cdots + a_{d_r} x^{d_r}$ [28].

multiSign$(\mathcal{SK}_i, \mathbb{V}, \mathfrak{msg})$: This algorithm requires several rounds of communication between signing parties (e.g., DO and SPs). A compact multi-signature σ is generated if all participants are honest, which means that the multiSign algorithm terminates immediately whenever one signer is dishonest. It takes as inputs a message $\mathfrak{msg}$, the current signer's signing key $\mathcal{SK}_i$, and a set of verification keys $\mathbb{V} := \{\mathcal{VK}_1, \mathcal{VK}_2, \cdots, \mathcal{VK}_n\}$ of all participants. The multi-signature $\sigma := (\mathcal{RC} \leftarrow \prod_{i=1}^n \mathcal{RC}_i, \mathcal{MS} \leftarrow \sum_{i=1}^n \mathcal{MS}_i \bmod p)$ is produced as a two-tuple containing the aggregated partial signatures $\mathcal{MS}$ and the nonce commitment $\mathcal{RC}$. It is generated based on the signing algorithm presented in Bellare and Neven's Multisig scheme [27], and the adapted version is shown in Fig. 5. In our system, two forms of data need to be multi-signed: the EHR blocks per se and the ciphertext containing the secret relationships between them. Hence, we define σ_{Φ_i} as $\sigma_{\Phi_i} \leftarrow multiSign(\mathcal{SK}_i, \mathbb{V}, \mathfrak{msg} = H(\Phi_i \| \mathfrak{N}_i \| \mathfrak{pp} \| t))$ to represent the multi-signature for a given EHR block, and we define σ_{E_i} as $\sigma_{E_i} \leftarrow multiSign(\mathcal{SK}_i, \mathbb{V}, \mathfrak{msg} = H(E_i \| \mathfrak{pp} \| t))$ to represent the multi-signature of the secret relationships. These ensure that DOs and SPs cannot refute their responsibility for the EHRs provided and allow TDB and DUs to verify the integrity and authenticity of the EHRs when necessary.

verify$(\sigma, \mathbb{V}, \mathfrak{msg})$: This deterministic algorithm is the last key algorithm in the ACCUMULATION phase. The TDB and DUs can execute it to verify the multi-signature σ. It starts by gathering the

challenge numbers: $ch_i \leftarrow H_1\left(\langle\mathbb{V}\rangle\|\mathcal{VK}_i\|\mathcal{RC}\|\mathfrak{msg}\right)$ for $\forall i \in \{1, 2, \cdots, n\}$ as in the third round of the signing process via an ideal cryptographic hash function $H_1 : \{0,1\}^* \rightarrow \{0,1\}^{m\in\mathbb{N}}$. These challenge numbers are then applied to the final validation expression: $g^{\mathcal{MS}} \stackrel{?}{=} \mathcal{RC} \prod_{i=1}^{n} \mathcal{VK}_i^{ch_i}$. According to MS-BN, the verification fails ($\perp \leftarrow \text{verify}(\sigma, \mathbb{V}, \mathfrak{msg})$) if the above equation does not hold. The whole ACCUMULATION phase will also fail, and the data cannot be stored at this point. Therefore, legitimate EHRs can only be saved to the TDB if all the accompanying signatures σ are validated by the TDB.

4.4. RETRIEVAL Phase

Once the DU confirms that the accompanying multi-signature is not a forgery, he/she can call the following algorithm to decrypt the ciphertext hierarchically. Please note that the higher the access rights represented by the DU's attributes, the larger the number of pointers that can be revealed.

decryptPointer($\mathfrak{pp},\mathbb{C},\mathcal{DK}_i$): It takes as input the public parameters $\mathfrak{pp}$, ciphertext $\mathbb{C}$, and the current decrypting entity's decryption key $\mathcal{DK}_i$. The inner ciphertext $\mathbb{C}_l$ can be decrypted. The secret pointer $\mathfrak{N}_l$ can be retrieved if the DU's attributes embedded in $\mathcal{DK}_i$ satisfy the patient-defined access structure $\mathbb{A}$, with respect to the connected sub-trees $\{\mathbb{A}_i''\}_l$ and the security level k_l. More concretely, each security level needs to be evaluated separately for obtaining different $\mathfrak{N}_i$. Following the authors of ML-ABE, their algorithm starts decrypting from the outer level using the decryption algorithm developed in the classic CP-ABE proposed by Bethencourt, Sahai and Waters. For the internal level, the DU would be able to extract the enciphering secret $e(g,g)^{\mathfrak{r}\varsigma_l}$ from n_l identified sub-trees $\{\mathbb{A}_i''\}_l$ rooted at the root node if the k_l-security level is satisfied [28]:

$$F_{R_{k_l}} = \prod_{x\in\{\mathbb{A}_i''\}_l} e(g,g)^{\mathfrak{r}q_{parent(x)}(index(x))}$$

$$= e(g,g)^{\Sigma_{x\in\{\mathbb{A}_i''\}_l}\mathfrak{r}q_{parent(x)}(index(x))} = e(g,g)^{\mathfrak{r}\varsigma_l}$$

, where the function *parent(x)* is called to find the parent node of node x in $\mathbb{A}$. The index related to node x is located by calling the function *index(x)*. The secret $e(g,g)^{\mathfrak{r}\varsigma_l}$ can be used to derive a pointer $\mathfrak{N}_i$ that has been flagged with the specified security level. Having the secret key of the corresponding pointer extracted by a legitimate DU through the above steps, the pointer $\mathfrak{N}_i$ used to locate the corresponding EHR block can be obtained in its plaintext form by:

$$\frac{\tilde{\mathbb{C}}_{k_l}}{e\left(\mathbb{C}_{k_l}, \mathcal{D}\right)/F_{R_{k_l}}} = \frac{\mathfrak{N}_i \cdot e(g,g)^{\gamma\varsigma_l}}{e\left(g^{\delta\varsigma_l}, g^{(\gamma+\mathfrak{r})/\delta}\right)/e(g,g)^{\mathfrak{r}\varsigma_l}}$$

$$= \frac{\mathfrak{N}_i \cdot e(g,g)^{\gamma\varsigma_l}}{e(g,g)^{\gamma\varsigma_l}} = \mathfrak{N}_i$$

4.5. Working Principle of TDB

In this subsection, we explain the working principle of the TDB that forms one of the key components in the proposed E-Tenon system. As seen visually in the left part of Fig. 6, the TDB comprises several open tables and one secret table. The open table has three columns per row: pointer, EHR block and multi-signature. It is worth noting that all encrypted data are separated from the open table. This is
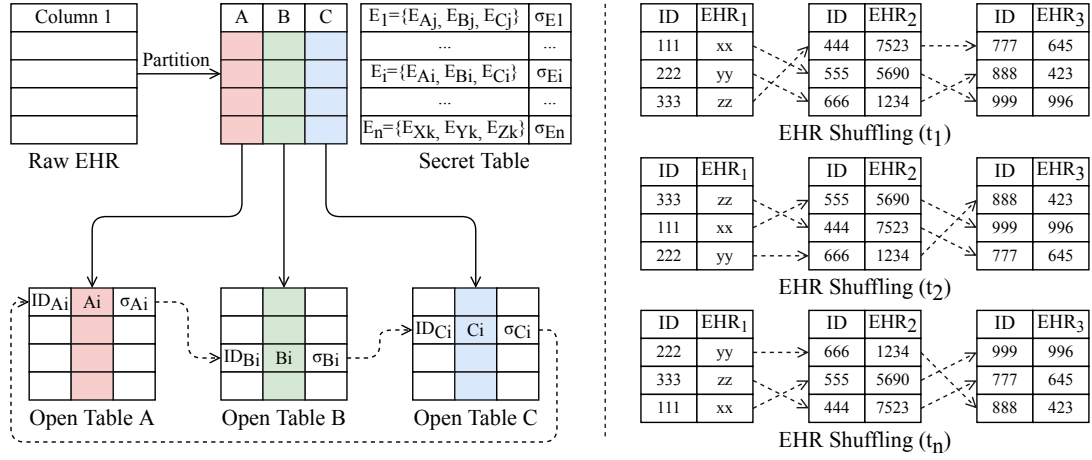
Fig. 6. Example of working principle of the Tenon database.

because we have adopted a multi-level ABE that produces a ciphertext containing multiple encrypted pointers. To reconstruct the data in the open tables, the authorised DU first decrypts the outer layer of the ciphertext. If successful, they will be presented with a series of encrypted pointers, and the number of pointers that can be decrypted depends on the DU's attributes. In this context, each row in the open table should not contain any encrypted pointers because this compromises the data confidentiality once a low privileged DU decrypts the outer ciphertext. Namely, an adversary can effortlessly use the encrypted pointers to locate the rows containing these pointers in the misconfigured open tables and directly combine them without the need to decrypt the secret pointers according to his/her attributes. Therefore, we collectively store all secret pointers accompanied by their multi-signature in a protected table isolated from other public tables. A legitimate DU can only read the entries that he/she is granted access to read. Moreover, the malicious outsider will not be able to see all the encrypted pointers and the malicious insider who can decrypt the outer layer of ciphertext will not be able to exploit the internal encrypted pointers to infer any information in the TDB.

Besides, we propose a complementary shuffling mechanism to reduce further the risk of any entity learning any information from the open data stored in the TDB. As demonstrated in the right part of Fig. 6, the TDB constantly shuffles the data to ensure that the order of the data is different each time the user accesses the TDB. Nevertheless, there is a possibility that the order of the data remains unchanged after the shuffle. If such a corner case occurs, the TDB will be automatically re-shuffled. This can be achieved by running a deterministic algorithm that compares the hash of the current data order with the hash of the previous data order. The algorithm returns $\perp$ when the shuffled data order is accidentally the same as the original data order. Thus, the TDB needs to re-shuffle the data to avoid this problem. These will further enhance the security of TDB and leave attackers with no rules to follow.

4.6. Signing Process

We use multi-signature to place constraints between the SP and the DO. This allows the DO to confirm that the EHR obtained from the SP is valid. On the other side, the SP can ensure that the DO has not attempted to alter the original EHRs they provided. It is therefore possible to guarantee the integrity and authenticity of the EHR if they have agreed to sign together on the *same* message.

The following describes two issues we need to address when signing. Firstly, imagine a signature obtained by encrypting the hash of a message generated via a one-way hash function. This signature is said to be valid if the hash value generated by the verifier using the same hash function on the accompanying message is equivalent to the hash obtained by decrypting the signature provided by the signer. Such a signing and verification process establishes the integrity of the message but does not maintain its confidentiality since the message used to generate the hash is in its original form [32]. The second issue is how the SP and DO sign the same content when there are inconsistencies between the data held by the SP and DO after preprocessing the EHRs. To address these issues, we propose the following steps for signers to securely multi-sign the same content. A visualisation of the process is provided (see Fig. 7).

- **Step 1**: DO calls dataPreprocessing(Φ) to preprocess EHRs and encryptPointer($\mathfrak{pp}$,$\mathbb{M}$,$\mathbb{A}$,$\{k_l\}$) to encrypt the pointers with self-defined access policies.
- **Step 2**: DO sends the preprocessed EHRs with encrypted pointers to SP.
- **Step 3**: SP decrypts all encrypted pointers using decryptPointer($\mathfrak{pp}$,$\mathbb{C}$,$\mathcal{DK}_i$) and reconstructs the data by joining EHR blocks in the right order. When DO allows legitimate SPs with authorised attributes to decrypt all secrets, there should be no concern since the original data comes from the SP.
- **Step 4**: SP compares the reconstructed data with the original data maintained by itself. If they are identical, then the SP and DO have agreed that the preprocessed EHRs have not been tampered with by the DO.
- **Step 5**: SP and DO interactively sign, using the algorithm multiSign($\mathcal{SK}_i$,$\mathbb{V}$,$\mathfrak{msg}$), on the hash of the confirmed EHR data obtained in step 2.

5. Security Analysis

In this section, we analyse and prove the security of our proposed scheme formally against the adversarial model described in Section 3. To ensure that E-Tenon is secure and resilient to a range of possible attacks, ML-ABE (a variant of CP-ABE) and MS-BN (a variant of Schnorr signature) are selected and integrated for reliability and validity. First, we note that ML-ABE is a proven CCA-1 secure scheme, where CCA-1 refers to the non-adaptive chosen-ciphertext attacks. Second, MS-BN is a proven secure scheme against the multi-user unforgeability against chosen message attacks (MU-UF-CMA). Our E-Tenon scheme should naturally inherit the security properties of these two building blocks.

Theorem 1. *Assume that the ML-ABE scheme in [28] is selectively CCA-1 secure. Then, the E-Tenon system preserves confidentiality and is selectively CCA-1 secure with respect to the CCA-1 security game and Definition 7.*

Proof. To prove the security of the E-Tenon system with respect to Definition 7, we consider there exist two polynomial-time adversaries $\mathcal{A}$ and $\mathcal{B}$, and a challenger $\mathcal{C}$. Here $\mathcal{B}$ is a simulator algorithm to run the security game defined in the naive CP-ABE. The security game $\mathcal{G}_{\mathcal{A}}^{CCA-1}(\lambda)$ is simulated as a non-adaptive chosen ciphertext attack against the proposed model by the adversary $\mathcal{A}$. It proceeds with $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ in four phases as follows:
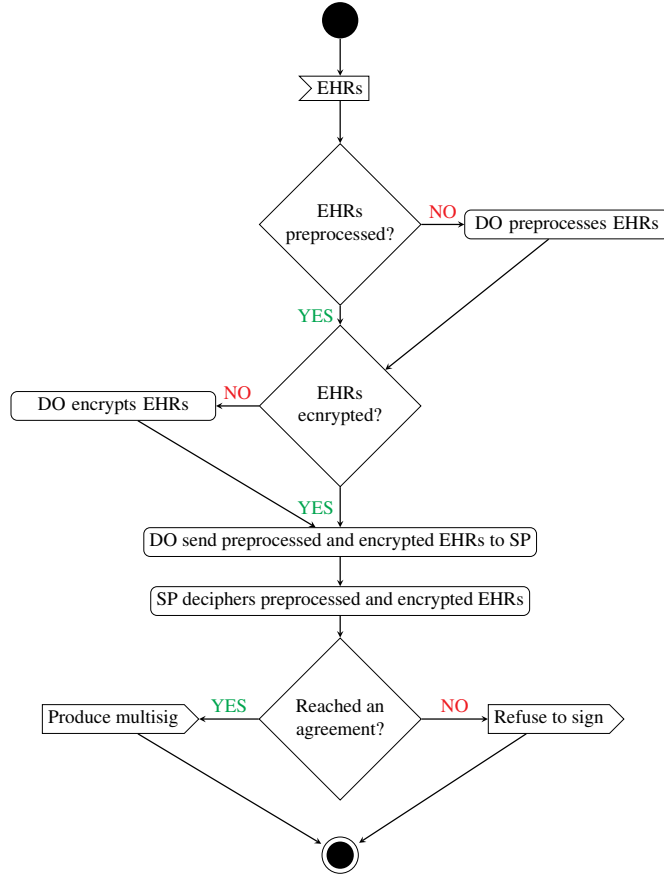
Fig. 7. How data owners and service providers can regulate each other to ensure the accuracy and integrity of EHRs.

- **Setup:** $\mathcal{C}$ runs setup algorithm with the security parameter λ to obtain the public parameters and the master key $\mathfrak{msk}, \mathfrak{pp} \leftarrow \text{setup}(\lambda)$, where $\mathfrak{msk}$ is defined as (δ, g^γ) and $\mathfrak{pp}$ is defined as $\{\mathbb{G}_0, \mathbb{G}_1, p, g, g^\delta, e, e(g, g)^\gamma\}$. Upon generation, $\mathcal{C}$ sends $\mathfrak{pp}$ to $\mathcal{B}$. Then $\mathcal{B}$ forward the same $\mathfrak{pp}$ to $\mathcal{A}$.

- **Query:** $\mathcal{B}$ initialises an empty table T, an integer session counter j starting from zero and an empty set $\mathbb{Q}$. $\mathcal{A}$ can repeatedly query the following during this phase:
  * **Create:** $\mathcal{B}$ asks $\mathcal{C}$ to increment j by 1. $\mathcal{B}$ asks $\mathcal{C}$ to run the setup algorithm $\mathfrak{msk}, \mathfrak{pp} \leftarrow \text{setup}(\lambda)$ and the keyGeneration algorithm $keys[\mathcal{DK}] \leftarrow \text{keyGeneration}(\mathfrak{pp}, \mathfrak{msk}, \mathbb{S})$ to extract a decryption key $\mathcal{DK}$ on $\mathbb{S}$ and the corresponding security levels k_l. Upon receiving $\mathcal{DK}$ from $\mathcal{C}$, $\mathcal{B}$ stores the entry $(j, \mathbb{S}, \mathfrak{pp}, \mathfrak{msk}, \mathcal{DK})$ in T if it is not a duplicate entry and shares the decryption key $\mathcal{DK}$ with $\mathcal{A}$.
  * **Corrupt:** $\mathcal{A}$ requests the decryption output of a ciphertext $\mathbb{C}$ using $\mathcal{DK}$ on $\mathbb{S}$. $\mathcal{B}$ checks if there is a previously extracted $\mathcal{DK}$ for $\mathbb{S}$ in the table T. If yes, $\mathcal{B}$ sets $\mathbb{Q} = \mathbb{Q} \cup \mathbb{S}$ and proceeds. Otherwise, $\mathcal{B}$ asks $\mathcal{C}$ to run the Create phase again and extract the corresponding $\mathcal{DK}$, such that the challenge access structure $\mathbb{A}^*(j, \mathbb{S}, k_l)$ is equal to 1.
  * **Decrypt:** Upon receiving $\mathcal{DK}$, $\mathcal{B}$ decrypts the ciphertext $\mathbb{C}$ with $\mathcal{DK}$ using the decryption algorithm presented in the naive CP-ABE scheme. Finally, $\mathcal{B}$ returns the decryption output of the ciphertext $\mathbb{C}$ to $\mathcal{A}$.

- **Challenge:** $\mathcal{A}$ chooses two plaintext messages $\mathbb{M}_0$ and $\mathbb{M}_1$ of the same length to be encrypted, which must remain unqueried until then. $\mathcal{A}$ also submits a challenge access structure $\mathbb{A}^*$ such that $\mathbb{S}$ does not satisfy $\mathbb{A}^*$ for all $\mathbb{S} \in \mathbb{Q}$. Upon receiving $\mathbb{A}^*$, $\mathcal{B}$ creates its own access structure $\mathbb{A}_{\mathcal{B}}$ based on the challenge access structure submitted by $\mathcal{A}$, such that $\mathbb{A}_{\mathcal{B}} \subseteq \mathbb{A}^*$. Next, $\mathcal{B}$ asks $\mathcal{C}$ to generate the ciphertext based on $\mathbb{M}_0$, $\mathbb{M}_1$ and $\mathbb{A}_{\mathcal{B}}$. $\mathbb{C}$ then randomly selects a bit $b \in \{0, 1\}$ and outputs the encryption results of $\mathbb{M}_b$ under $\mathbb{A}_{\mathcal{B}}$ to $\mathcal{B}$. Finally, $\mathcal{B}$ forwards the output to $\mathcal{A}$.
- **Guess:** $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$ for b. $\mathcal{A}$ wins the game if $b' = b$.

In order to determine the adversary's advantage at this stage, some basic observations are necessary to be made. It is noted that the element $\tilde{\mathbb{C}}_{k_l}$ within the ciphertext encrypted by $\mathcal{C}$ during the challenge phase is either $\mathbb{M}_0 \cdot e(g, g)^{\gamma\varsigma_l}$ or $\mathbb{M}_0 \cdot e(g, g)^{\gamma\varsigma_l}$. Thus, the advantage for the adversary to distinguish between the two cases is $\text{Adv}_{\mathcal{A}}^{CCA-1}(1^\lambda) \leqslant \epsilon$. Now, let us take into account a modified game $\mathcal{G}_{\mathcal{A}}^{CCA-1'}$. In this game, the main difference is that the element $\tilde{\mathbb{C}}_{k_l}$ of the challenge ciphertext becomes either $\mathbb{M}_0 \cdot e(g, g)^{\gamma\varsigma_l}$ or $\mathbb{M}_1 \cdot e(g, g)^{\theta}$, where θ is chosen at random out of an additive group, $\theta \xleftarrow{\$} \mathbb{O}_p$. Accordingly, the advantage of the adversary in winning the modified game becomes $\text{Adv}_{\mathcal{A}}^{CCA-1'}(1^\lambda) \geqslant \frac{1}{2} \cdot \epsilon$. Then we simulate the attack over the modified security game based on case 1 of [28]. A challenger $\mathcal{C}$ first chooses two exponents γ and δ at random from $\mathbb{Z}_p$, such that $\gamma, \delta \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{C}$ then obtains and shares the public parameters with the adversary in a special encoding: $\mathfrak{E}_0(1) = g$, $\mathfrak{E}_0(\delta) = g^\delta$ and $\mathfrak{E}_T(\gamma)$. In the subsequent challenge phase, the adversary $\mathcal{A}$ again asks challenger $\mathcal{C}$ to encrypt the challenge message under the access structure $\mathbb{A}'^*$. After that, the adversary $\mathcal{A}$ gets $\mathbb{C}_{k_l} = g^{\delta\varsigma_l}$ and $\tilde{\mathbb{C}}_{k_l} = e(g^\delta, g^\delta)^{\theta_l}$ for each defined security level along with the relevant attributes. It is worth pointing out that the request from adversary $\mathcal{A}$ will not be granted if $\mathcal{A}$ requests a set of attributes that can satisfy all the security levels defined in the challenge access structure. In other contradictory cases, the game terminates immediately, and the adversary loses the game. Finally, we use the big-$\mathcal{O}$ notation to express the upper limit of the adversary's advantage in winning the aforementioned security game as $\text{Adv}_{\mathcal{A}}^{CCA-1'}(1^\lambda) \leqslant \mathcal{O}(\frac{c^* \cdot q^2}{p})$, where c^* is the bound on the maximum number of security level can be set, q is the bound on the maximum number of group elements obtained by $\mathcal{A}$, and p is the order of an additive group $\mathbb{O}_p$. Hence, we state that the proposed E-Tenon system is CCA-1 secure and the confidentiality of EHR is guaranteed under the Generic Group Model if no PPT adversary can selectively break the security naive CP-ABE and ML-ABE with non-negligible advantage. $\square$

Theorem 2. *Assume that the ML-ABE scheme in [28] is private against both malicious and honest-but-curious adversaries. Then, the proposed E-Tenon system preserves privacy against both malicious DU and honest-but-curious TDB.*

Proof. In this proof, we consider attacks from a malicious DU and an honest-but-curious TDB, respectively. First of all, it is worth noting that the malicious adversary DU will have the same advantage as in $\mathcal{G}_{\mathcal{A}}^{CCA-1}(\lambda)$ when a DU tries to extend or override his/her access rights to gain additional access to the encrypted information (e.g., the embedded enciphering secret ς_l). This is because such a scenario is in line with the confidentiality property. Next, let us recall that the secret relationships $\{\mathfrak{N}_l^*\}_{l \in \{1, c^*\}}$ in the ciphertext are independently encrypted with a set of different security levels $\{k_l^*\}_{l \in \{1, c^*\}}$ thanks to the use of multi-level ABE. Thus, in order to deduce any information from any part of a challenge ciphertext, or to break the indistinguishability property, the adversary DU must be able to recover $e(g, g)^{\gamma\varsigma_l}$ together with the corresponding $\tilde{\mathbb{C}}_{k_l} = \mathfrak{N}_i \cdot e(g, g)^{\gamma\varsigma_l}$ and $\mathcal{D} = g^{\frac{\gamma + \mathfrak{r}}{\delta}}$. However, the proof of Theorem 1 shows that the adversary only has a negligible advantage in selectively breaking the CCA-1 security of E-Tenon.

Our framework, therefore, prevents malicious DUs from revealing any information, as ML-ABE does not disclose any useful information.

In another scenario, let us assume that the honest-but-curious TDB complies with its obligations. However, it tries to reveal which DO upload the EHR or which DU requested to retrieve the EHR. This clearly compromises the privacy property. Having said that, we show that the TDB does not have the ability to distinguish requesters by their attributes. Suppose DO_x and DO_y are two patients with a set of distinct attributes in the proposed system. Their $\mathbb{A}$ will be indistinguishable as ML-ABE inherits such property from the naive CP-ABE scheme, such that $\mathbb{A}(\mathbb{S}_{DO_x}) = 1$ and $\mathbb{A}(\mathbb{S}_{DO_y}) = 1$ for $\mathbb{S}_{DO_x} \neq \mathbb{S}_{DO_y}$. Therefore, the honest-but-curious TDB is unable to identify DOs and DUs. Hence, our system is secure against both internally and externally launched attacks. $\quad\square$

Theorem 3. *Assume that the MS-BN scheme in [27] is MU-UF-CMA secure. Then the proposed E-Tenon system is MU-UF-CMA secure with respect to the MU-UF-CMA security game and Definition 8.*

Proof. Let $\mathcal{F}$ be a PPT adversary running in time at most t against the multi-signature algorithm. Let q_p and N denote the number of signing processes initiated by $\mathcal{F}$ and the number of verification keys in the set $\mathbb{V}$, respectively, and let q_r be the maximum number of random oracle queries that $\mathcal{F}$ can make.

As proved in [27], breaking the MS-BN model is considered to be at least as hard as the discrete logarithm problem (DLP) for an adversary $\mathcal{F}$ under the random oracle model (ROM). Below we recapitulate several important points discussed by Bellare and Neven based on their Forking Lemmas. Firstly, the accepting probability acc and the forking probability frk of $\mathcal{F}$ used in their General Forking Lemma are quantified as follows:

$$\text{frk} \geqslant \text{acc} \cdot \left(\frac{\text{acc}}{q} - \frac{1}{h} \right)$$

$$\text{acc} \geqslant \epsilon - \frac{(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{2q_p (q_r + N \cdot q_p)}{2^k}$$

Then, square of the acceptance rate acc gives us the acc^2 as below:

$$\text{acc}^2 \geqslant \left(\epsilon - \frac{(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{2q_p(q_r + N \cdot q_p)}{2^k} \right)^2$$

$$\geqslant \epsilon^2 - \frac{\epsilon(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{\epsilon \cdot 2q_p(q_r + N \cdot q_p)}{2^k}$$

$$- \frac{\epsilon(q_r + N \cdot q_p + 1)^2}{2^{l_0}} + \frac{(q_r + N \cdot q_p + 1)^4}{(2^{l_0})^2}$$

$$+ \frac{(q_r + N \cdot q_p + 1)^2}{2^{l_0}} \cdot \frac{2q_p(q_r + N \cdot q_p)}{2^k}$$

$$- \frac{\epsilon \cdot 2q_p(q_r + N \cdot q_p)}{2^k} + \left(\frac{2q_p(q_r + N \cdot q_p)}{2^k} \right)^2$$

$$+ \frac{2q_p(q_r + N \cdot q_p)}{2^k} \cdot \frac{(q_r + N \cdot q_p + 1)^2}{2^{l_0}}$$

$$\geqslant \epsilon^2 - \frac{2\epsilon(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{4\epsilon \cdot q_p(q_r + N \cdot q_p)}{2^k}$$

$$\geqslant \epsilon^2 - \frac{2(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{4q_p(q_r + N \cdot q_p)}{2^k}$$

If there exists an adversary $\mathcal{F}$ who manages to win the game $\mathcal{G}_{\mathcal{F}}^{ROM}(t, q_p, q_r, N, \epsilon)$, then it implies that there is an adversary $\mathcal{F}'(\epsilon', t')$ that can solve the DLP. Thus, the probability ϵ' of adversary $\mathcal{F}'$ successfully solving the DLP and the corresponding running time t' for $\mathcal{F}'$ to solve the DLP are given by:

$$t' = 2t + q_p t_{\exp} + \mathcal{O}\left((q_p + q_r)\left(1 + q_r + Nq_p\right)\right)$$

$$\epsilon' \geqslant \mathrm{frk}$$

$$\geqslant \mathrm{acc} \cdot \left(\frac{\mathrm{acc}}{q} - \frac{1}{h}\right)$$

$$\geqslant \frac{\mathrm{acc}^2}{q} - \frac{\mathrm{acc}}{h}$$

$$\geqslant \frac{\mathrm{acc}^2}{q} - \frac{1}{2^{l_1}}$$

$$\geqslant \frac{\epsilon^2 - \frac{2(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{4q_p(q_r + N \cdot q_p)}{2^k}}{q_r + q_p} - \frac{1}{2^{l_1}}$$

$$\geqslant \frac{\epsilon^2}{q_r + q_p} - \frac{2q_r + 16N^2 \cdot q_p}{2^{l_0}} - \frac{8N \cdot q_p}{2^k} - \frac{1}{2^{l_1}}$$

Here, t' is two times the running time t required by $\mathcal{F}$ plus the time required to solve the DLP. One can argue that if there is no algorithm capable of solving the DLP. Then there is no adversary capable of breaking the security of MS-BN with any reasonable probability. Therefore, the proposed E-Tenon system is also MU-UF-CMA secure against integrity and authenticity attacks by inheriting the security properties of the Multi-Signature scheme MS-BN. □

6. Performance Evaluation

In this section, we discuss the performance of the proposed model. We first compare our scheme with other competitive solutions in terms of security properties. We then evaluate the relevant computation cost of the E-Tenon in different tasks. Subsequently, we discuss the communication and storage costs of E-Tenon.

6.1. Security Properties

To compare security properties and functionalities, we have selected several state-of-the-art schemes ([15, 16, 18, 33–35]) for protecting EHRs and compared them on various dimensions. A summarised comparison of the security properties and characteristics of the schemes is presented in Table 2. Although

Table 2

Security properties and functionalities comparison with related works.

	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
[18]	✗	✓	✓	✗	✗	✓	✓	✗	✓	◗
[15]	✗	✓	✓	✓	✓	◗	✓	✗	✓	✗
[33]	✗	✓	✓	✓	✓	✓	-	✗	✗	✗
[34]	✗	✗	✓	✗	✗	✓	◗	✗	✓	◗
[16]	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓
[30]	✗	✗	✓	✓	✓	✗	✗	-	-	-
[35]	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Fully Satisfied   ✗: Not Satisfied   ◗: Partially Satisfied   -: N/A

SP1: Open Database; **SP2**: Secure-channel Free; **SP3**: Data Confidentiality;
SP4: Data Integrity; **SP5**: Non-Repudiation; **SP6**: User Privacy;
SP7: Collusion Resistance; **SP8**: Multi-level Access Control;
SP9: Fine-grained Access Control; **SP10**: Process Transparency

there are wide-ranging interesting solutions, they still suffer from different shortcomings and do not work efficiently where open databases are concerned. The scheme proposed by Sun et al. [15] employs attribute-based techniques, but the patient's involvement in the encryption and signing of the data is weakened. In [15], the patient does not have the right to specify the access policy of their own data. Further, the doctor is responsible for encrypting and signing the data, meaning that the doctor has direct control over the data, rather than the patient. Such a design increases the advantage for malicious insiders and makes the system less trustworthy for patients. In contrast, our E-Tenon system inherently gives more control to the patients since they are the actual owner of the EHR. In this way, they can set different levels of access policies for different types of data on their own, and they are allowed to engage in the process of Multi-Signature.

Green et al. [18] have attempted to reduce the user's computational overheads by outsourcing the task of decryption to an untrusted cloud service provider (CSP). In their system, the CSP transforms ABE's ciphertext into a simple El Gamal-style ciphertext based on a transformation key provided by the data user. Despite the converted ciphertext requiring lower computational cost than its initial form when recovering the plaintext, the user cannot verify that the CSP has performed the transformation operation honestly. In [30]] the author(s) proposed three different signature schemes, those can ensure confidentiality, integrity and non-repudiation services. Similarly, the scheme presented in [33] ensures unlinkability of the stored data by converting identifying attributes into non-sensitive pseudonyms. However, this process is not transparent, meaning the data owner cannot audit their data flow. By comparison, the data pre-processing algorithm in our system is run on the data owner's side, and there is no need for other central entities to perform any secondary processing of the uploaded EHRs. Besides, instead of using a basic form of digital signature, we utilise multi-signature technology, which allows a group of participants to co-sign the same message effectively. This naturally enables the patient (DO) and the service provider (SP) to restrain each other's dishonest behaviour. Thus, it further enhances integrity, authenticity and non-repudiation. In this regard, we emphasise that multi-signature is more promising than the standard digital signature or other techniques that involve many signatures. Because in the absence of multiple entities constraining each other, the entity accessing the EHR later can replace the EHR provided by the previous entity and continue to sign the EHR supplied by itself. Therefore, the conventional signature approach does not guarantee the authenticity of the EHR in a collaborative environment.

Table 3

Performance benchmarking in terms of computation, communication and storage cost at DO, DU, SP, TDB's side.

Computational Cost	E-Tenon			Zhang et al.[16]			Alexandra[30]		
Dataset		MIMIC-III	eICU		MIMIC-III	eICU		MIMIC-III	eICU
Signing Cost (at DO, SP)	$\mathcal{T}_{\mathbf{exp}}$	$\approx 2.34\ ms$	$\approx 2.34\ ms$	$3\mathcal{T}_{\mathbf{mult}}$	$\approx 43.50\ ms$	$\approx 43.50\ ms$	$\mathcal{T}_{\mathbf{exp}}$	$\approx 2.34\ ms$	$\approx 2.34\ ms$
Verification Cost (at DU, TDB)	$\mathcal{T}_{\mathbf{exp}}$	$\approx 2.34\ ms$	$\approx 2.34\ ms$	$\mathcal{T}_{\mathbf{mult}} + 3\mathcal{T}_{\mathbf{par}}$	$\approx 25.84\ ms$	$\approx 25.84\ ms$	$\mathcal{T}_{\mathbf{mult}} + \mathcal{T}_{\mathbf{par}}$	$\approx 18.28\ ms$	$\approx 18.28\ ms$
Encryption Cost (at DO)	$k\mathcal{T}_{\mathbf{mult}} + 2(k + l_{MST})\mathcal{T}_{\mathbf{exp}}$	$\approx 142.70\ ms$	$\approx 208.34\ ms$	$k\mathcal{T}_{\mathbf{mult}} + 2k(1 + l_{AT})\mathcal{T}_{\mathbf{exp}}$	$\approx 329.90\ ms$	$\approx 1627.32\ ms$	–	–	–
Decryption Cost (at SP, DU)	$(\mathrm{n}_{MST} + l_{AT})(2\mathcal{T}_{\mathbf{par}} + \mathcal{T}_{\mathbf{exp}} + \mathcal{T}_{\mathbf{mult}}) + \mathcal{T}_{\mathbf{mult}}(2 + m\mathrm{n}_{MST}) + \mathcal{T}_{\mathbf{par}}$	$\approx 761.28\ ms$	$\approx 1094.46\ ms$	$(k\mathrm{n}_{AT} + l_{AT})(2\mathcal{T}_{\mathbf{par}} + \mathcal{T}_{\mathbf{exp}} + \mathcal{T}_{\mathbf{mult}}) + \mathcal{T}_{\mathbf{mult}}(2 + m\mathrm{n}_{AT}) + \mathcal{T}_{\mathbf{par}}$	$\approx 1249.28\ ms$	$\approx 5854.16\ ms$	–	–	–
Communication & Storage Cost	E-Tenon			Zhang et al.[16]			Alexandra[30]		
Dataset		MIMIC-III	eICU		MIMIC-III	eICU		MIMIC-III	eICU
Signature (at DO, DU, SP, TDB)	$2\|ecc\|$	$\approx 320\ bits$	$\approx 320\ bits$	$3\|ecc\|$	$\approx 480\ bits$	$\approx 480\ bits$	$2\|ecc\|$	$\approx 320\ bits$	$\approx 320\ bits$
Ciphertext (at DO, DU, SP, TDB)	$\{\|MST\|, 2(k + l_{MST})\|\mathbb{G}\|\}$	$\approx 3.86\ kb$	$\approx 4.59\ kb$	$\{k\|AT\|, 2k(1 + l_{AT})\|\mathbb{G}\|\}$	$\approx 14.18\ kb$	$\approx 56.72\ kb$	–	–	–

$\mathcal{T}_{\mathbf{exp}}$: cost of a modular exponentiation (2.34 ms); $\mathcal{T}_{\mathbf{mult}}$: cost of a multiplication (14.5 ms); $\mathcal{T}_{\mathbf{par}}$: cost of a bilinear pairing (3.78 ms);
l: number of external nodes - i.e. attributes in the tree (10); n: number of internal nodes - i.e. threshold gates in the tree (5); $|ecc|$: size of
the elliptic curve (160 bits); m: number of child nodes of the threshold gates (5); $|MST|$: size of an aggregate access tree (160 bits);
$|AT|$: size of a separate access tree (160 bits); k: number of access tree (5); $|\mathbb{G}|$: bit length of the element in the group (1024 bits)

(a) Signing

(b) Verification

(c) Encryption

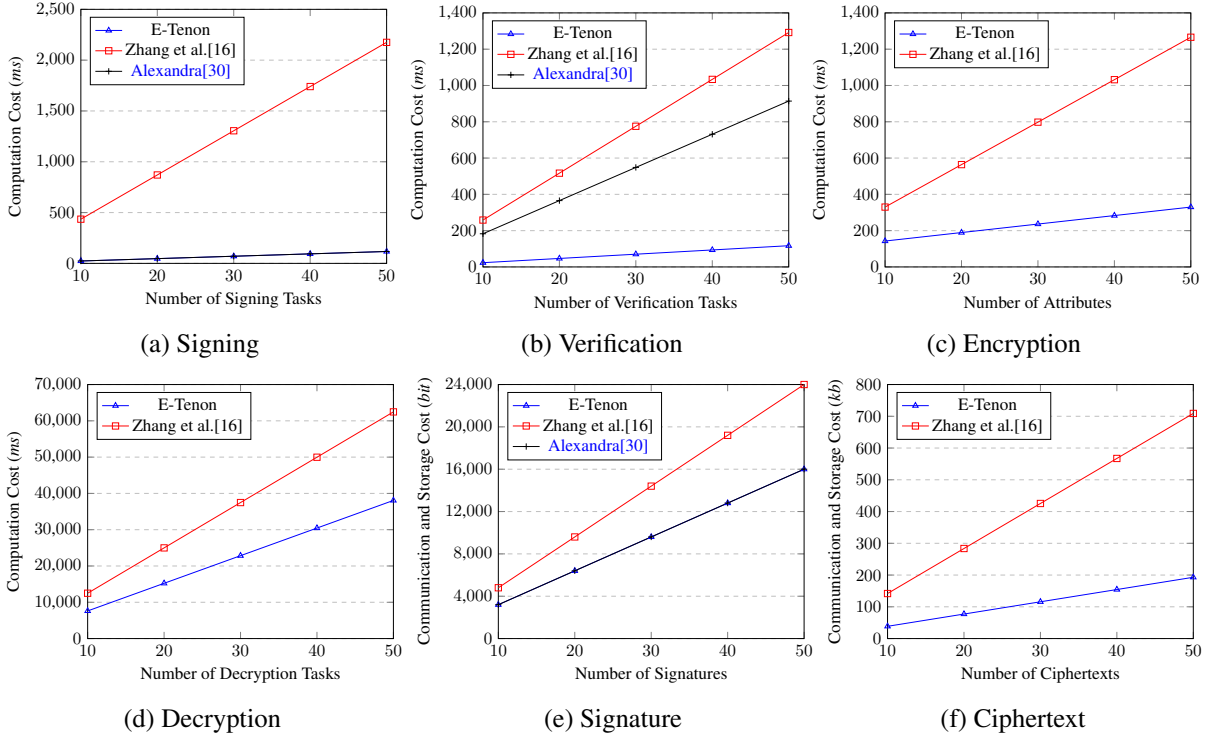(d) Decryption

(e) Signature

(f) Ciphertext

Fig. 8. Performance comparison based on computation, communication and storage cost of MIMIC-III dataset (using simulation parameters specified in Table 3). Note that in figure (a): Signing and figure (e): Signature, there is some overlapping between two different results due to the same operation time.

Furthermore, Huang et al.'s solution [34] focuses specifically on EHR confidentiality, although their solution is not security channel-free and we found no discussion of how they ensure EHR integrity, which makes their solution slightly less than perfect in our comparison. However, system proposed by Zhang et al. [16] (SSH) and system proposed by Maffei et al. [35] (GORAM) satisfied most of the security properties. GORAM allows data owners to share their data stored in the cloud selectively, and the storing entity is not permitted to inspect any data. Nevertheless, their robust security comes at the cost of increasing the ciphertext size and slowing down encryption and decryption.
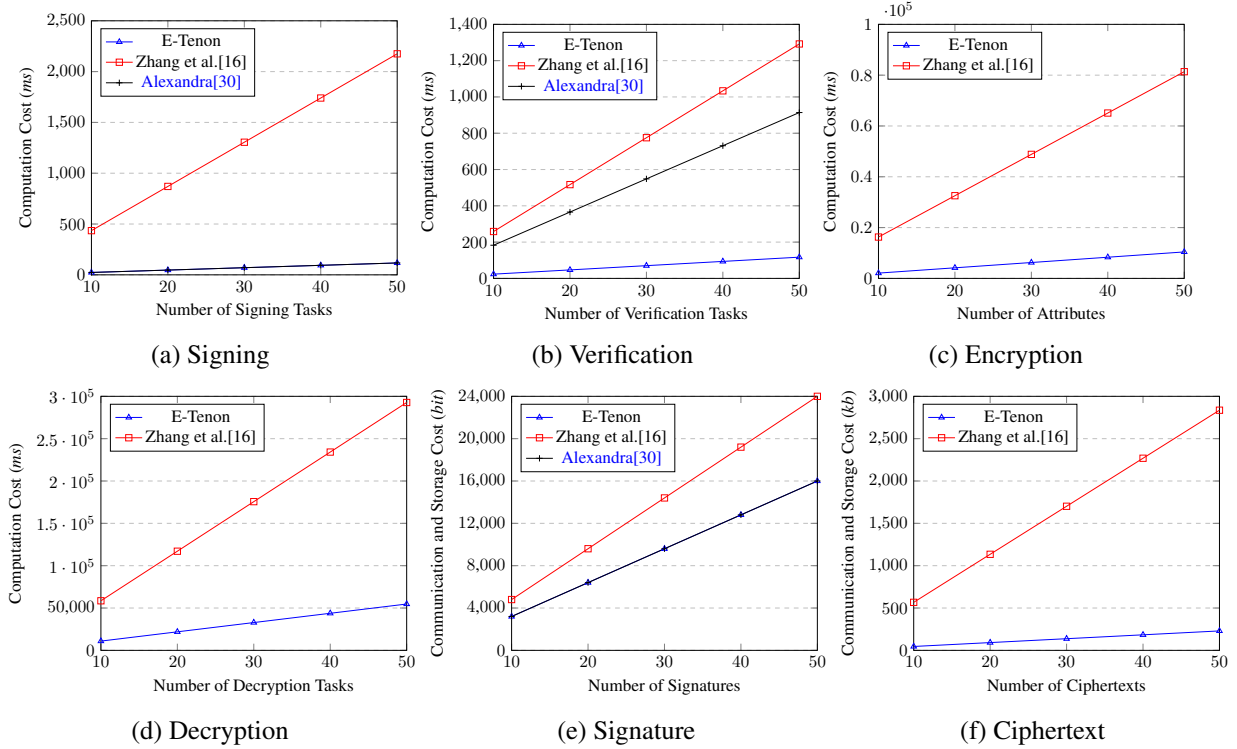
Fig. 9. Performance comparison based on computation, communication and storage cost of eICU dataset (using simulation parameters specified in Table 3). Note that in figure (a): Signing and figure (e): Signature, there is some overlapping between two different results due to the same operation time.

Finally, we observe that none of those mentioned above schemes can be applied to public databases where most of the data is stored in plaintext, and none of the encryption methods used in these schemes can efficiently implement multi-level access control. On the contrary, thanks to the novel concept of E-Tenon, our data is securely stored in an open database (TDB), which means the computational overhead on encryption and decryption is minimal compared to solutions based on heavy encryption.

6.2. Computation Cost

We use a virtual machine (Ubuntu 12.04) with an Intel Core i5-4200M dual-core 2.50 GHz CPU to conduct simulations of the core operations based on three main libraries: JPBC library Pbc-05.14 [36], JCE library [37] and Apache Commons IO library [31]. We test modular exponentiation, multiplication and bilinear pairing 2,000 times and take the average CPU time in milliseconds. Regarding the dataset, we used the MIMIC-III v1.3 dataset [38], a freely accessible healthcare database offered by the Massachusetts Institute of Technology. This dataset contains de-identified electronic health records (EHRs) of over 30,000 unique patients. However, since the dataset does not contain any sensitive or personally identifiable information such as names, addresses, phone numbers, or emails, we considered a number of the randomly selected columns to be sensitive attributes/identifiers for our experiments. This approach allowed us to achieve our experimental objectives without modifying the original dataset.

We also analyze the performance of the proposed scheme based on another common dataset: the eICU Collaborative Research Database, which consists of 31 Tables, each containing 8-10 columns. Since our proposed scheme does not encrypt the whole dataset, it hides the relationship among the data, which depends on the number of sensitive attributes present in each table. Our analysis shows that the proposed performance is quite similar to the MIMIC-III v1.3 dataset (as shown in Fig. 9).

In conducting the comparison, we found it difficult to find schemes with similar security properties and performance metrics, especially in an open database environment, for a fully fair comparison. We acknowledge that the system proposed by Zhang et al. [16] (SSH) and the system proposed by Maffei et al. [35] (GORAM) have similar security properties to ours and could be considered candidates for comparison. However, we would like to point out that [16] mainly relies on aggregate signature and an anonymous CP-ABE technique, while [35] (GORAM) does not use similar attribute-based techniques and signature schemes but instead uses batched zero-knowledge proofs of shuffle and an accountability technique based on chameleon signatures. Our scheme, on the other hand, is based on multi-signature and multi-level CP-ABE. Therefore, a focused comparison with [16] would be fair since our scheme and [16] share similar techniques, although the details of our schemes differ significantly.

Table 3 shows the cost at data owner, data user, service provider and database side for signing, verification, and encryption and decryption. Firstly, the signing and verification algorithms adapted in our model outperform other relevant algorithms in the state-of-the-art schemes [16, 30, 39]. This is because only one exponentiation operation is required when an entity signs/verifies the message (the average CPU time for 2000 trials is approximately equal to only 2.34 *ms*). In addition, since it is a practical requirement to protect different types of EHR data according to different levels of security, our system uses ML-ABE's aggregated master access structure to meet this requirement effectively. It is worth noting that the schemes built on the classic CP-ABE (e.g. [16]) need to create a separate access structure for each defined security level $\{k_l\}_{l \in \{1,c\}}$ in order to achieve the same security functionality as we have. However, using multiple access structures will inevitably create many duplicate attributes. So our system saves computational overhead by avoiding duplicate nodes and unnecessary polynomials in the access structure, such that $\sum_{l=1}^{c} \mathfrak{l}_{AT_{k_l}} \geqslant \mathfrak{l}_{MST}$ ($\mathfrak{l}$ denotes the number of attributes/external nodes). A more intuitive comparison of performance with [16] is visualised in Fig. 8. Furthermore, the advantages of our approach can also be seen in the following scenario. It is common knowledge that the size of EHR can vary from a few bits to tens or even hundreds of megabytes (e.g., 100 bits - 100 MB). However, we are only encrypting relationships between different EHR blocks; instead of encrypting the whole EHR data, we are only encrypting some constant-sized pointers (16 Bytes). This idea considerably reduces the time taken for encryption and decryption, thanks to the use of the *electronic tenon structure*.

6.3. Communication and Storage Cost

Finally, we analyse the communication and storage costs of the proposed protocol. As mentioned above, the access structure used by E-Tenon is designed in an aggregated manner, and the cost of our scheme in terms of communication and storage is optimised by eliminating duplicate attributes. This implies that the ciphertext size in the E-Tenon system is shorter than other schemes with a series of separate access structures. However, our protocol requires an extra round of communication during the signing process compared to other schemes, a trade-off for supporting concurrent signing in the multi-user environment, as pointed out in MS-BN [27]. That being said, the size of our signature is only $2|ecc|$ (note that different schemes may work over a different n-bit elliptic curve), the size of the public key and system parameters for a single signer is 320 bits only as a whole. Following the security

discussion in [40], the use of a 160-bit elliptic curve would provide about the equivalent security level as DSA (Digital Signature Algorithm) and RSA (Rivest–Shamir–Adleman) with a 1024-bit modulus. Therefore, let us assume that we currently require the same level of security as stated above. In this case, the multi-signature σ is only 320 bits (40 Bytes). As a data owner/user, the communication cost depends on the size of the ciphertext, the size of the plaintext, the size of the multi-signature and the size of the keys. The size of the ciphertext depends on the number of attributes, the length of attribute names, and the number of access trees. In our experiments, the number of attributes assigned is 10, the size of a single plaintext message is 128 bits, the size of an aggregate access tree $|MST|$ is 160 bits, the size of the multi-signature σ is 320 bits, and the size of the keys is 1024 bits. Thus, a total of 1760 (220 bytes) has to be transmitted. As a service provider, the communication costs are similar to those of the data owner/user, except that they are responsible for uploading EHRs to the database, which involves extra communication costs depending on the size of non-PII data and the encrypted data pointers. Taken together, the discussion suggests that we have achieved more secure and reliable protection of EHR without compromising efficiency.

7. Conclusion and Future Work

This paper proposed an efficient privacy-preserving open data-sharing scheme for a secure EHR system. The idea of keeping most of the data open without compromising security and privacy is considered a novel attempt in this field. Moreover, we presented in detail the effective integration of two promising technologies in our E-Tenon system: ML-ABE and Multi-Signature, to protect the security of EHR and patient privacy. Our solution exploits the advantages of ABE for key management and multiple signatures for protecting the authenticity and integrity of EHR. The multi-level security supported by ML-ABE allows us to protect the relationships between EHR blocks independently with different levels of security, where only legitimate DU with appropriate attributes can decrypt a certain number of pointers and sensibly join the open data. These not only improve the security of EHR but also grant patients the ability to share EHRs efficiently. In addition, with the formal security analysis, our solutions have been proven to be capable of preventing a range of possible security attacks. Finally, we have analysed the costs and performance of the E-Tenon system in various aspects. The simulation results show that our E-Tenon system does not compromise security properties while maintaining promising efficiency and flexibility.

Future Work. However, adapting our current generic system for EHRs data-sharing may face one challenge, particularly in dealing with the Break-the-Glass (BtG) [41, 42] situations. Since we give more priority to EHR data security and patient privacy, where entities with valid security attributes can only access patient data, this may cause difficulty in supporting BtG situations where a system can compromise to provide emergency access to the patient's data. Therefore, in our future work, we aim to address this in an effective way and explore the use of Fog Computing technology to enhance our solutions in a way that can further improve efficiency (e.g., by outsourcing part of the EHR preprocessing and decryption tasks to edge devices) while maintaining strong security.

References

[1] J. Bethencourt, A. Sahai and B. Waters, Ciphertext-policy attribute-based encryption, in: *2007 IEEE Symposium on Security and Privacy (SP '07)*, IEEE, 2007, pp. 321–334. doi:10.1109/SP.2007.11.

[2] V. Goyal, O. Pandey, A. Sahai and B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: *Proceedings of the 13th ACM conference on Computer and communications security - CCS '06*, ACM Press, 2006, pp. 89–98. doi:10.1145/1180405.1180418.

[3] S. Belguith, N. Kaaniche and M. Hammoudeh, Analysis of attribute-based cryptographic techniques and their application to protect cloud services, *Transactions on Emerging Telecommunications Technologies* (2019). doi:10.1002/ett.3667.

[4] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma and L. Wei, Auditable σ-time outsourced attribute-based encryption for access control in cloud computing, *IEEE Transactions on Information Forensics and Security* **13**(1) (2018), 94–105. doi:10.1109/tifs.2017.2738601.

[5] P. Gope and T. Hwang, BSN-care: A secure IoT-based modern healthcare system using body sensor network, *IEEE Sensors Journal* **16**(5) (2016), 1368–1376. doi:10.1109/jsen.2015.2502401.

[6] F. Rezaeibagha and Y. Mu, Distributed clinical data sharing via dynamic access-control policy transformation, *International Journal of Medical Informatics* **89** (2016), 25–31. doi:10.1016/j.ijmedinf.2016.02.002.

[7] A. Bahga and V.K. Madisetti, A cloud-based approach for interoperable electronic health records (EHRs), *IEEE Journal of Biomedical and Health Informatics* **17**(5) (2013), 894–906. doi:10.1109/jbhi.2013.2257818.

[8] X. Liu and X. Yi, Privacy-preserving collaborative medical time series analysis based on dynamic time warping, in: *Computer Security – ESORICS 2019*, Springer International Publishing, 2019, pp. 439–460. ISBN 978-3-030-29962-0. doi:10.1007/978-3-030-29962-0_21.

[9] S. Xu, J. Ning, Y. Li, Y. Zhang, G. Xu, X. Huang and R. Deng, A secure EMR sharing system with tamper resistance and expressive access control, *IEEE Transactions on Dependable and Secure Computing* (2021), 1–1. doi:10.1109/tdsc.2021.3126532.

[10] H. Wang, J. Ning, X. Huang, G. Wei, G.S. Poh and X. Liu, Secure fine-grained encrypted keyword search for e-healthcare cloud, *IEEE Transactions on Dependable and Secure Computing* (2019), 1–1. doi:10.1109/tdsc.2019.2916569.

[11] J. Shi, J. Lai, Y. Li, R.H. Deng and J. Weng, Authorized keyword search on encrypted data, in: *Computer Security - ESORICS 2014*, Springer International Publishing, 2014, pp. 419–435. doi:10.1007/978-3-319-11203-9_24.

[12] M. Li, S. Yu, Y. Zheng, K. Ren and W. Lou, Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption, *IEEE transactions on parallel and distributed systems* **24**(1) (2012), 131–143. doi:10.1109/TPDS.2012.97.

[13] S. Xu, J. Ning, X. Huang, Y. Li and G. Xu, Untouchable once revoking: A practical and secure dynamic EHR sharing system via cloud, *IEEE Transactions on Dependable and Secure Computing* (2021), 1–1. doi:10.1109/tdsc.2021.3106393.

[14] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai and R. Attia, Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT, *Computer Networks* **133** (2018), 141–156. doi:10.1016/j.comnet.2018.01.036.

[15] Y. Sun, R. Zhang, X. Wang, K. Gao and L. Liu, A decentralizing attribute-based signature for healthcare blockchain, in: *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, 2018. doi:10.1109/icccn.2018.8487349.

[16] Y. Zhang, R.H. Deng, G. Han and D. Zheng, Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things, *Journal of Network and Computer Applications* **123** (2018), 89–100. doi:10.1016/j.jnca.2018.09.005.

[17] A. Kumari, V. Kumar, M.Y. Abbasi, S. Kumari, P. Chaudhary and C.-M. Chen, CSEF: Cloud-based secure and efficient framework for smart medical system using ECC, *IEEE Access* **8** (2020), 107838–107852.

[18] M. Green, S. Hohenberger and B. Waters, Outsourcing the decryption of ABE ciphertexts, in: *Proceedings of the 20th USENIX Conference on Security (SEC'11)*, USENIX Association, USA, 2011, p. 34. doi:10.5555/2028067.2028101.

[19] G.G. Dagher, J. Mohler, M. Milojkovic and P.B. Marella, Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology, *Sustainable Cities and Society* **39** (2018), 283–297. doi:10.1016/j.scs.2018.02.014.

[20] N. Li, T. Li and S. Venkatasubramanian, t-Closeness: Privacy beyond k-Anonymity and l-Diversity, in: *2007 IEEE 23rd International Conference on Data Engineering*, 2007, pp. 106–115. doi:10.1109/ICDE.2007.367856.

[21] S. Brands, *Rethinking public key infrastructures and digital certificates: building in privacy*, Mit Press, 2000.

[22] W. Lueks, G. Alpár, J.-H. Hoepman and P. Vullers, Fast revocation of attribute-based credentials for both users and verifiers, *computers & security* **67** (2017), 308–323.

[23] L. SWEENEY, k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **10**(05) (2002), 557–570. doi:10.1142/s0218488502001648.

[24] A. Machanavajjhala, D. Kifer, J. Gehrke and M. Venkitasubramaniam, L-Diversity: Privacy beyond k-Anonymity, *ACM Trans. Knowl. Discov. Data* **1**(1) (2007), 3–es. doi:10.1145/1217299.1217302.

[25] H. Wang and Y. Song, Secure Cloud-Based EHR System Using attribute-based cryptosystem and blockchain, *Journal of Medical Systems* **42**(8) (2018). doi:10.1007/s10916-018-0994-6.

[26] C.P. Schnorr, Efficient signature generation by smart cards, *Journal of Cryptology* **4**(3) (1991), 161–174. doi:10.1007/bf00196725.

[27] M. Bellare and G. Neven, Multi-Signatures in the plain public-key model and a general forking lemma, in: *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, Association for Computing Machinery, New York, NY, USA, 2006, pp. 390–399–. ISBN 1595935185. doi:10.1145/1180405.1180453.

[28] N. Kaaniche and M. Laurent, Attribute based encryption for multi-level access control policies, in: *SECRYPT 2017: 14th International Conference on Security and Cryptography*, Vol. 6, Scitepress, 2017, pp. 67–78.

[29] J. Li, X. Chen, J. Li, C. Jia, J. Ma and W. Lou, Fine-grained access control system based on outsourced attribute-based encryption, in: *Computer Security – ESORICS 2013*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 592–609. ISBN 978-3-642-40203-6. doi:10.1007/978-3-642-40203-6_33.

[30] A. Boldyreva, Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme, in: *International Workshop on Public Key Cryptography*, Springer, 2003, pp. 31–46. doi:10.1007/3-540-36288-6_3.

[31] The Apache Software Foundation, Apache Commons IO - UUID generation libraries. https://commons.apache.org/sandbox/commons-id/uuid.html.

[32] J.H. An, Y. Dodis and T. Rabin, On the security of joint signature and encryption, in: *Advances in Cryptology — EUROCRYPT 2002*, L.R. Knudsen, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 83–107. ISBN 978-3-540-46035-0. doi:10.1007/3-540-46035-7_6.

[33] J. Camenisch and A. Lehmann, (Un)Linkable pseudonyms for governmental databases, in: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1467–1479–. ISBN 9781450338325. doi:10.1145/2810103.2813658.

[34] C. Huang, R. Lu, H. Zhu, J. Shao and X. Lin, FSSR: Fine-grained EHRs sharing via similarity-based recommendation in cloud-assisted EHealthcare system, in: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 95–106–. ISBN 9781450342339. doi:10.1145/2897845.2897870.

[35] M. Maffei, G. Malavolta, M. Reinert and D. Schroder, Privacy and access control for outsourced personal records, in: *2015 IEEE Symposium on Security and Privacy*, IEEE, 2015. doi:10.1109/SP.2015.28.

[36] B. Lynn, Pbc library. https://crypto.stanford.edu/pbc/download.html.

[37] Oracle Technology Network, Java cryptography extension (JCE). https://www.oracle.com/java/technologies/javase-jce-all-downloads.html.

[38] A.E.W. Johnson, T.J. Pollard, L. Shen, L.-w.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi and R.G. Mark, MIMIC-III, a freely accessible critical care database, *Scientific Data* **3**(1) (2016). doi:10.1038/sdata.2016.35.

[39] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham and B. Waters, Sequential aggregate signatures and multisignatures without random oracles, in: *Advances in Cryptology - EUROCRYPT 2006*, Springer Berlin Heidelberg, 2006, pp. 465–485. doi:10.1007/11761679_28.

[40] N. Koblitz, A. Menezes and S. Vanstone, The state of elliptic curve cryptography, *Designs, Codes and Cryptography* **19**(2/3) (2000), 173–193.

[41] A.D. Brucker and H. Petritsch, Extending Access Control Models with Break-Glass, in: *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies*, SACMAT '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 197–206–. ISBN 9781605585376. doi:10.1145/1542207.1542239.

[42] S. Vavilis, M. Petković and N. Zannone, A severity-based quantification of data leakages in database systems, *Journal of Computer Security* **24**(3) (2016), 321–345.