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

Zhihui Lin[1], Prosanta Gope[1(✉)], Jianting Ning[2], and Biplab Sikdar[3]

[1] Department of Computer Science,
University of Sheffield, Sheffield S1 4DP, UK
zhihuilin111@gmail.com, p.gope@sheffield.ac.uk
[2] College of Computer and Cyber Security,
Fujian Normal University, Fuzhou 350117, China
jtning88@gmail.com
[3] Department of Electrical and Computer Engineering,
National University of Singapore, Singapore 117583, Singapore
bsikdar@nus.edu.sg

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 are excessively encrypting the whole database, where for each access request the entire database is required to be decrypted, which is a time-consuming process. On the other hand, the use of EHR for medical research (e.g. development of precision medicine, diagnostics techniques etc.) as well optimisation of practises in healthcare organisations, requires the EHR to be analysed and for that they should be easily accessible without compromising the privacy of the patient. In this paper, we propose an efficient technique called E-Tenon that not only securely keeps all EHR publicly accessible but also provides the desirable 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 concept of E-Tenon empowers patients to securely share their EHR under multi-level, fine-grained access policies defined by themselves. Analyses show that our system outperforms existing solutions in terms of computational complexity.

Keywords: open database · e-tenon · ABE · multi-signature.

1 Introduction

With the rapid development of Health Information Technology (HIT) and cloud services, a growing number of 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 patient's consent and typically include a range of sensitive and private information such as patient's identity codes, health history, medical diagnoses and treatment plans [25]. Obviously, leaking these data can cause embarrassment or even life-threatening consequences to patients. Indeed, in reality, despite record levels of security spending by different hospitals, there are still a wide range of malicious cyber attacks intended to penetrate databases and connected systems. This is because cybercriminals find EHR highly profitable, which stimulates them to steal such data by various means. Therefore, the need to design a system that preserves patient privacy in a robust and efficient manner is imperative. In particular, many traditional schemes are vulnerable and ineffective. For example, many security researchers recommend strict encryption of databases so that the data is protected to the greatest extent possible, even in a security incident. However, it is inefficient to excessively encrypt the whole database or a majority of the data as it will have a marked impact on the performance of the database. Besides, EHRs are increasingly being used for developing customised and precision medicine regimens, developing new and more accurate techniques for diagnosis and treatment, and optimise 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.

Related Work. Since medical data security has become a growing public concern, a considerable number of schemes have been published so far for secure medical data sharing and privacy preservation [20,2,15,22,14,3,23,26,12]. For instance, most research in protecting medical data have emphasised the use of cryptographic methods such as Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Key-Policy Attribute-Based Encryption (KP-ABE) [3,26,9]. However, none of them has considered the use of secure open databases to save avoidable overheads on encryption and decryption. System architecture proposed in [20] 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 [14], 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 don't know who will be accessing their data, as mentioned in [5]. In addition, Belguith et al. [3] proposed a multi-authority CP-ABE scheme that delegates expensive computing tasks to cloud servers, and their scheme also prevents collusion between the authorities. Although there are wide-ranging interesting solutions [23,9,7,18], they still suffer from different shortcomings. We observe that although the scheme proposed by Sun et al. [23] employs attribute-based techniques, the patient's involvement in the encryption and signing of the data is weakened. In [23], the patient does not have the right to specify the access structure. Also, the doctor handles the encryption and signing process, meaning that the direct control of the data is entirely in the hands of the doctor 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 would naturally give more control to the patients since they are the ac-

tual 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. [9] 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 transform the ciphertext of ABE into a simple ElGamal-style ciphertext based on a transformation key provided by the data user. Despite the converted ciphertext requires less computational cost than its initial form when recovering the plaintext, the user cannot verify that the CSP has performed the transformation operation honestly. Similarly, the scheme presented in [7] 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 the flow of their data. Another system named GORAM proposed by Maffei et al. [18] satisfies 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, the strong security they have achieved comes at the cost of increasing the ciphertext size and slowing down encryption and decryption.

Our Motivation and Contribution. It is unfortunate that most of the existing solutions in the literature require excessive encryption, and there is no work which is able to ensure patient's privacy while keeping data open. Likewise, as argued in [8], excessive security may obstruct sensible data use by healthcare providers and patients, and most approaches have failed to properly weigh the patient's right to privacy against the legitimate sharing of data. As previously mentioned, a promising cryptographic technique for encrypting data at a fine granularity is ABE. Although several similar works have used ABE to protect EHR, they use it to encrypt the entire database, which is computationally intensive. 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, we found that many schemes failed to make proper use of digital signatures to ensure data integrity and authenticity. For example, [26] allows only one entity to sign the EHR, which grants the entity too much power. Although some schemes allow multiple entities to sign the data, they cannot guarantee that all participants will sign the same content. 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 and propose 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 protection of data does not depend on heavy encryption and decryption. Rather, the protection of EHRs is achieved through the data preprocessing, maintenance of secret relationships between EHR blocks and shuffling techniques. Notably,

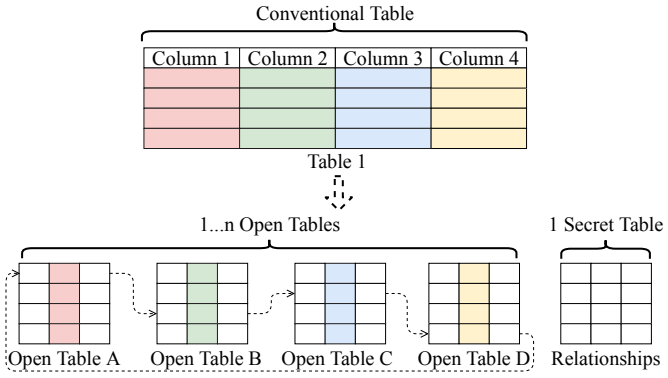

Fig. 1. Overview of the proposed tenon database. A conventional table will be segmented into a series sub-tables where the relationship between rows are hidden. It can be revealed partially or fully depending on the data user's attributes (access rights).

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 is constantly shuffled, which makes it extremely difficult for attackers to exploit open data. The main contributions of this paper are summarised as follows:

- We design an efficient open database in which most of the data is open, and only a minor portion needs to be encrypted. Thus, it requires less computation than other schemes in terms of encrypting and decrypting data.
- We integrate and extend Multi-Signature and Multi-level Attribute-based Encryption techniques to satisfy all desired security properties.
- We present data preprocessing and shuffling methods used in conjunction with the proposed E-Tenon system to securely store and share EHRs.
- We show how to ensure that multiple entities sign the same content, even after preprocessing. 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 integrity and authenticity of the EHR. It takes only 2.34 ms in signing and verifying the signature, 19.18 ms and 57.18 ms in encryption and decryption, 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 the focus of this work on EHR, the concept of E-Tenon would also be applicable in other scenarios which require low-latency access to user data, such as in mobile edge computing environments.

Roadmap. The rest of the paper is organised as follows. Section 2 introduces and recapitulates the required mathematical notations, security assumptions and related schemes. In Section 3, we present 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 cost analysis in 5 and 6, respectively. Section 7 of the paper concludes our work in light of all that has been mentioned.

2 Preliminaries

In this section, we introduce and recapitulate several prerequisites, including definitions of some mathematical notations, Kaaniche et al.'s ML-ABE scheme [10], and Bellare et al.'s Multi-Signature scheme (BN-MS)[4].

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 $\mathbb{G}^*$.

Buiding Blocks. More formal definitions are provided below. Bilinear maps are a useful tool of pairing-based cryptography because they are convenient to 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 [21,4]). *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 (Bilinear Maps [5]). *Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of same prime order p. g is an arbitrary generator $g \xleftarrow{\$} \mathbb{G}_0$. e is a symmetric bilinear map, such that $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ where $e(g^x, g^y) = e(g^y, g^x) = e(g, g)^{xy} = e(g, g)^{yx}$.*

The security of the E-Tenon system we proposed will be based on the Discrete Logarithm Assumption. The assumption holds when $Adv^{dlog}_{\mathbb{G}_0}(\mathcal{A})$ is negligible.

Definition 3 (Discrete Logarithm Assumption [4]). *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 $\mathbb{G}_0$:*

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

Multi-level CP-ABE. The multi-layered and intertwined doctor-patient relationships across different healthcare providers make it impractical to protect EHRs. Each distinct part of the EHR file may require to be accessed with completely different access rights depending on the purpose of data user. Therefore, the naive CP-ABE is not fully compatible in our scenario. However, as one of the successors to CP-ABE, ML-ABE fills in the gaps.

Definition 4 (ML-ABE [10]). *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* $\mathbb{M} = \{m_l\}_{l\in\{1,c\}}$ *with respect to 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 provide the definition of 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* $\mathcal{DK}$*. Note that* $\mathbb{C}$ *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 [10]). *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* $\mathbb{A}'_x(\mathbb{S}) = \perp$*.*

Multi-Signature. A Multi-Signature (MS) solution allows a group of signers to co-sign on a common document in a compact manner [4]. As a real-life example, the publication of a report/document often requires the cooperation of multiple colleagues. In order to guarantee the authenticity of the information in the report, each participant needs to sign the file. Therefore, Multi-Signature technology are 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 [13]. Thus, we prefer to use Multi-Signature scheme that are not based on comparatively more burdensome requirements of PKI (e.g., knowledge of secret key hypothesis [6]) to enhance the practicality of the proposed E-Tenon system further. Bellare and Neven's MS-BN [4] defined below fits well with our concept.

Definition 6 (MS-BN [4]). *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* $\mathcal{SK} := r \xleftarrow{\$} \mathbb{Z}_p$ *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 important facts about MS-BN. First, the security of this scheme is guaranteed on the assumption that at least one of the signers is honest [4, 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 breach of security when all the signers are honest-but-curious or dishonest. In view of the increasing sophistication of cyber attacks, any end-user can no longer be undoubtedly trusted. Hence, our model will strengthen MS-BN to accommodate the case 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.

3 System and Adversarial Model

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 [4] with an encryption scheme ML-ABE [10]. 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: Verifier (VER).

- **CTA** is a fully trusted entity responsible for generating system-wide public parameters for all participants within the system.

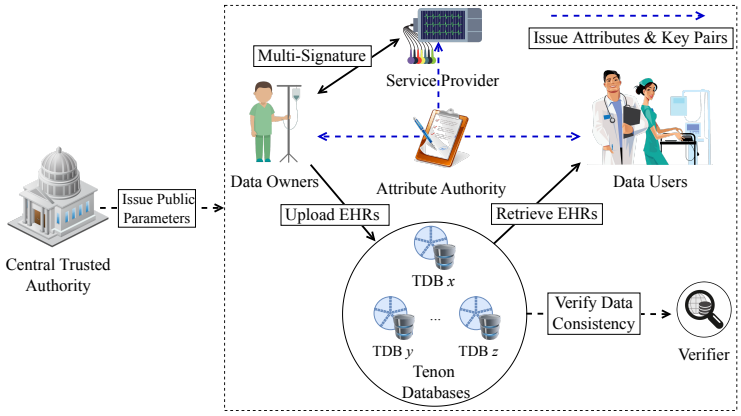

Fig. 2. System model of the proposed scheme.

- **AA** serves in a similar way as the CTA. It is in charge of the management of the user's attributes and the issuance of secret keys for the user, where appropriate. We note that the state-of-the-art multi-authority ABE systems use several different AAs and make each AA responsible for only one specific attribute. However, it must resist collusion attacks. In our case, we do not require multiple AAs, and we consider the AA as a trusted entity.
- **DO** is the actual owner of the EHRs, i.e., the patient. Normally, DOs are concerned about the privacy of their EHRs and they have the right to control the sharing of their EHRs. However, DO can also be malicious, for example, DO 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 the data is sent to the database. DOs will also be required to multi-sign their data.
- **SP** is an honest-but-curious entity involved in the signing process. It provides unconfirmed EHR to the DO. For example, a smart blood pressure sensor provides readings to the DO (such data remain subject to patient confirmation). However, one exceptional SP who can be trusted is the patient's doctor in charge (they provide patients with officially confirmed diagnostic results and treatment plans).
- **TDB** is an honest-but-curious entity responsible for the data management. TDB per se is a distributed open database. The data should be stored as it is, and TDB has no right to decrypt any of the 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.

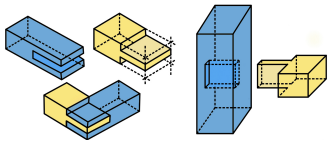

Fig. 3. An example of mortise and tenon joints.

- **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 the different attributes they hold. Moreover, a DU without the required attributes will be considered malicious when attempting to decrypt the secret pointers.
- **VER** is a trusted participant who is responsible for auditing data consistency between multiple TDBs. The synchronisation of EHR across multiple databases enhances data availability and avoids single points of failure.

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 greatly increase the applicability and the efficiency of the model, it also exposes system interactions and EHRs in transit to a variety of 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 the secret relationships.
- **Privacy Threat**: DO and DU's identity may be revealed when they interact with a semi-trusted TDB. A malicious DU may be able to 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, it remains a challenge to ensure the integrity and authenticity of the EHR provided by patients. 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.

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 a 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 $\mathcal{A}$.
- **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 $\mathbb{S} \in \mathbb{Q}$. $\mathbb{C}$ 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 a 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 a 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*

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

4 Concrete Construction

Overview. 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. Table 1 lists some essential notations and cryptographic functions we used. The proposed E-Tenon system benefits from the effective integration of ML-ABE [10] and MS-BN [4]. We firstly state the innovations and extensions we have made to these building blocks. To the best of our knowledge, the existing ABE-based privacy-preserving systems pose many redundant encryption and decryption overheads. However, our solution confidently allows EHRs to be securely made open (without encryption) in the TDB after special preprocessing. In concrete terms, EHR blocks stored in the TDB can only be mapped into meaningful information by deciphering relevant secret pointers. In addition, data shuffling techniques are applied to constantly change the position and order of EHR blocks. This signifies that the open data is presented to DU at random each time the TDB is accessed. Furthermore, in the original MS-BN, there must be a trusted signing entity involved in the signing process, but we cannot assume that this will always be the case in a safety-critical application. Therefore, we do not necessarily need the presence of a fully trusted signer to ensure the unforgeability of a multi-signature, which makes our E-Tenon system is more flexible and robust. Eventually, we present steps grounded on sound logic to ensure that the SPs and DOs can always sign the same message. Taken together, these provide us with the ability to manage EHRs in an efficient, flexible and granular manner while maintaining privacy and security at the same time.

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
$\mathcal{SDK}$	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

Table 1. Notations and cryptographic functions.

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 type of EHR when each category of data contains at least κ different types.

– The diagnosis will only be provided to the patient after it has been confirmed by medical experts. Moreover, a doctor in charge, as a trusted SP, in an ideal state will not be bribed by anyone to provide fake diagnosis to the patient.

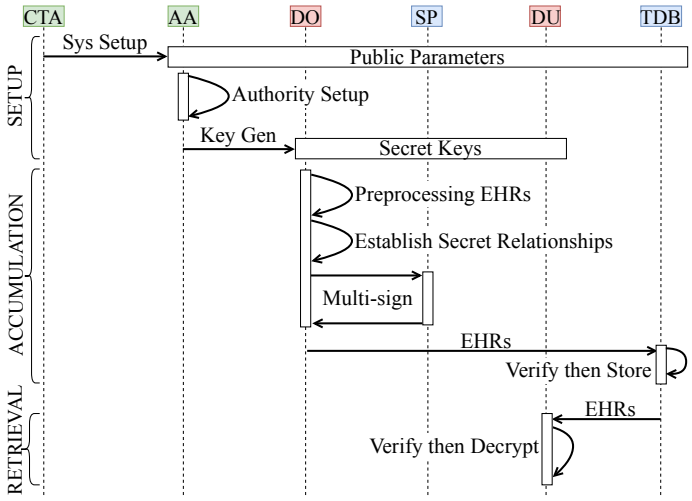

Fig. 4. Work flow of the proposed scheme.

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 may 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 can be established, it first needs to be classified into two main categories: identifiable data and Non-PII data. The EHR preprocessing algorithm will recommend patients to perform minor encryption on identifiable data as well as the secret relationships between EHR blocks. The Non-PII data classified by our algorithm will be made open after preprocessing as it can not be used to trace a patient's identity. Note that when encryption is performed with a patient-defined access policy, it is equivalent to the patient giving consent to those users who satisfy the access policy. Subsequently, the DO and SP multi-sign the data such that the TDB can refuse to store the data if the signature is found to be 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 DU also has the option to verify the signature of the data, and they can decrypt the pointers at different security levels according to their attributes when they believe that 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

Let λ be the implicit security parameter denotes the size of the cryptographic groups, and let $\mathbb{S} := \{a_1, a_2, \cdots, a_n\}$ be the universe of entity's attributes. The following two algorithms need to be administered by the CTA and AA to complete 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 element γ, 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 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 the data confidentiality by colluding. All necessary keys for the user to operate both ABE 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.

4.3 ACCUMULATION Phase

In order to understand what must be encrypted and what can be left open we need to consider the ways in which data may be combined. For instance, an insecure combination is the National Insurance Number (NINO) with the medical condition since it reveals 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 patient's identity (e.g., almost everyone may have a cough), detailed symptom information can be useful in inferring 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).

dataPreprocessing(Φ): This algorithm runs by the DO. It begins by classifying and labelling EHRs by identifiable and non personally identifiable information (Non-PII). As an example, identifiable columns may include patient's NINO, mobile number. Non-identifiable columns include medical condition, gender, symptom, blood pressure. Next, it splits any tokenisable and Non-PII EHRs

Algorithm 1 dataPreprocessing(Φ)

1: **Begin:**
2: **Step 1** classifies Φ by *identifiable* and *Non-PII*.
3: **Step 2** If $tokenisable(\Phi) == true$, splits Non-PII Φ into data blocks, each block contains one main word plus the preceding stopwords if present.
4: **Step 3** establishes the *relationships* between an identifiable column and Non-PII columns as well as the *relationships* between blocks of Non-PII columns.
5: **Step 4** returns preprocessed/structured EHRs $\mathbb{M}$.
6: **End.**

into blocks with the relationships between blocks 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 [24]. 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 EHRs 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.

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 [10] are adapted as below:

$$\mathbb{C} := \left\{\mathbb{A}, \forall k_l : \{\mathbb{A}_i''\}_l, \mathbb{C}_{k_l}, \tilde{\mathbb{C}}_{k_l}, \forall y : \mathbb{C}_y, \mathbb{C}_y'\right\}$$
$$\begin{cases}\mathbb{C}_{k_l} = g^{\delta\varsigma_l}, & \tilde{\mathbb{C}}_{k_l} = \mathfrak{N}_i \cdot e(g,g)^{\gamma\varsigma_l} \\ \mathbb{C}_y = g^{q_y(0)}, & \mathbb{C}_y' = H(\mathrm{att}(y))^{q_y(0)}\end{cases}$$

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) [5]. Here the enciphering secret ς_l embedded in each ciphertext with a particular security level k_l is computed as $\varsigma_l := \sum_{i\in\{1,2,\cdots,n_l\}} q_r\left(\mathrm{index}\left(x_i\right)\right)$ where $q_r(x)$ is the polynomial related to the root node r of $\mathbb{A}$, $q_r(x) = a_0 + a_1 x + \cdots + a_{d_r} x^{d_r}$ [10].

multiSign($\mathcal{SK}_i$,$\mathbb{V}$,$\mathfrak{msg}$): This algorithm requires several rounds of communication between sigining 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

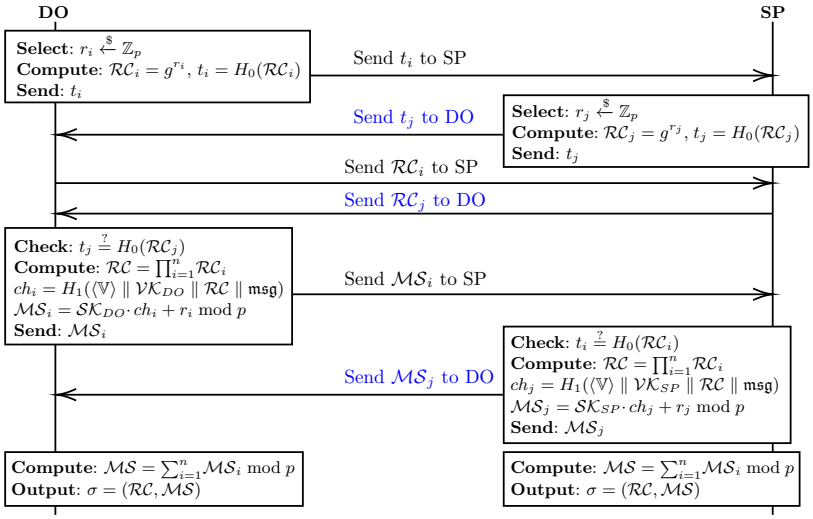

Fig. 5. Rounds of communication in multiSign algorithm.

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 [4], and the adapted version is shown in Fig. 5. In our system, there are two forms of data that 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(σ,$\mathbb{V}$,$\mathfrak{msg}$): This deterministic algorithm is the last key algorithm in the ACCUMULATION phase. It can be executed by the TDB and DU to verify the multi-signature σ. It starts by gathering the challenge numbers: $ch_i \leftarrow H_1(\langle \mathbb{V} \rangle \| \mathcal{VK}_i \| \mathcal{RC} \| \mathfrak{msg})$ 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.

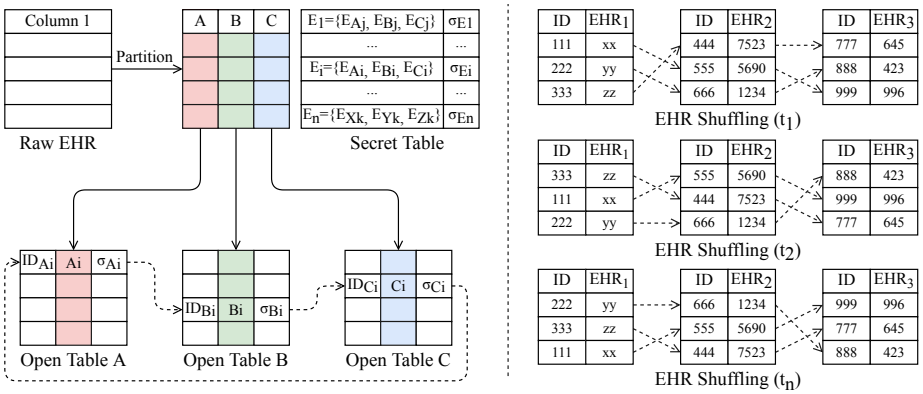

Fig. 6. Example of working principle of the Tenon database.

Please note that the higher the access rights represented by the DU's attributes, the higher 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 and 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-tress $\{\mathbb{A}_i''\}_l$ 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 [10]: $e(g,g)^{\mathfrak{r}\varsigma_l} = \prod_{x\in\{\mathbb{A}_i''\}_l} e(g,g)^{\mathfrak{r}q_{parent(x)}(index(x))} = e(g,g)^{\sum_{x\in\{\mathbb{A}_i''\}_l}\mathfrak{r}q_{parent(x)}(index(x))}$. 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 is composed of several open tables and one secret table. There are three columns per row in the open table: pointer, EHR block and multi-signature. It is worth noting that all encrypted data is separated from

the open table. This is 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 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 its multi-signature in a protected table isolated from other public tables. A legitimate DU can only read the entries that he/she is granted 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 exploit the internal encrypted pointers to infer any information in the TDB. Besides, we propose a complementary shuffling mechanism to further reduce 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 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 need to reshuffle 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 can sign together on the *same* message. The following describes two issues we need to address when signing. Firstly, imagine a signature that is 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 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 [1]. 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.

- **Step 1**: DO calls dataPreprocessing(Φ) and encryptPointer($\mathfrak{pp}$,$\mathbb{M}$,$\mathbb{A}$,$\{k_l\}$) to preprocess EHRs and 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. There should be no concern when DO allows legitimate SPs with authorised attributes to decrypt all secrets 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 can confirm that the preprocessed EHRs has 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 its 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 the ML-ABE scheme in [10] is selectively CCA-1 secure, then the proposed 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 $\mathcal{A}$, $\mathcal{B}$ 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 $\mathcal{A}$, $\mathcal{B}$,$\mathcal{C}$ in four phases: Setup, Query, Challenge and Guess as follows:

- **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:
  - **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 $keys[\mathcal{DK}] \leftarrow$ 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 that 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 message $\mathbb{M}_0$ 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, $\mathcal{B}$ forward 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 $Adv_{\mathcal{A}}^{CCA-1}(1^\lambda) \leq \epsilon$. 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 $Adv_{\mathcal{A}}^{CCA-1'}(1^\lambda) \geq \frac{1}{2} \cdot \epsilon$. Then we simulate the attack over the modified security game based on case 1 of [10]. 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 $Adv_{\mathcal{A}}^{CCA-1'}(1^\lambda) \leq \mathcal{O}(\frac{c^* \cdot q^2}{p})$, 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 a non-negligible advantage. □

Theorem 2. *Assume the ML-ABE scheme in [10] 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 a honest-but-curious TDB, respectively. First of all, it is worthy noticeable that the malicious adversary DU will have the same advantage as in $\mathcal{G}_{\mathcal{A}}^{CCA-1}(\lambda)$ when 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 uploaded 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. □

Theorem 3. *Assume the MS-BN scheme in [4] 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 [4], 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:

$$\begin{aligned}
\text{frk} &\geq \text{acc} \cdot \left(\frac{\text{acc}}{q} - \frac{1}{h} \right) \\
\text{acc} &\geq \epsilon - \frac{\left(q_r + N \cdot q_p + 1\right)^2}{2^{l_0}} - \frac{2q_p \left(q_r + N \cdot q_p\right)}{2^k}
\end{aligned}$$

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

$$\begin{aligned}
\text{acc}^2 &\geq \left(\epsilon - \frac{(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{2q_p(q_r + N \cdot q_p)}{2^k} \right)^2 \\
&\geq \epsilon^2 - \frac{\epsilon(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{\epsilon \cdot 2q_p(q_r + N \cdot q_p)}{2^k} \\
&\quad - \frac{\epsilon(q_r + N \cdot q_p + 1)^2}{2^{l_0}} + \frac{(q_r + N \cdot q_p + 1)^4}{\left(2^{l_0}\right)^2} \\
&\quad + \frac{(q_r + N \cdot q_p + 1)^2}{2^{l_0}} \cdot \frac{2q_p(q_r + N \cdot q_p)}{2^k} \\
&\quad - \frac{\epsilon \cdot 2q_p(q_r + N \cdot q_p)}{2^k} + \left(\frac{2q_p(q_r + N \cdot q_p)}{2^k} \right)^2 \\
&\quad + \frac{2q_p(q_r + N \cdot q_p)}{2^k} \cdot \frac{(q_r + N \cdot q_p + 1)^2}{2^{l_0}} \\
&\geq \epsilon^2 - \frac{2\epsilon(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{4\epsilon \cdot q_p(q_r + N \cdot q_p)}{2^k} \\
&\geq \epsilon^2 - \frac{2(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{4q_p(q_r + N \cdot q_p)}{2^k}
\end{aligned}$$

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:

$$\begin{aligned}
t' &= 2t + q_p t_{\text{exp}} + \mathcal{O}\left(\left(q_p + q_r\right)\left(1 + q_r + N q_p\right)\right) \\
\epsilon' &\geq \text{frk} \\
&\geq \text{acc} \cdot \left(\frac{\text{acc}}{q} - \frac{1}{h} \right) \\
&\geq \frac{\text{acc}^2}{q} - \frac{\text{acc}}{h} \\
&\geq \frac{\text{acc}^2}{q} - \frac{1}{2^{l_1}} \\
&\geq \frac{\epsilon^2 - \frac{2(q_r + N \cdot q_p + 1)^2}{2^{l_0}} - \frac{4q_p(q_r + N \cdot q_p)}{2^k}}{q_r + q_p} - \frac{1}{2^{l_1}} \\
&\geq \frac{\epsilon^2}{q_r + q_p} - \frac{2q_r + 16N^2 \cdot q_p}{2^{l_0}} - \frac{8N \cdot q_p}{2^k} - \frac{1}{2^{l_1}}
\end{aligned}$$

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 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 Cost Analysis

In this section, we intend to perform the cost analysis of the proposed model. We first evaluate the relevant computation cost of the E-Tenon in various aspects. Subsequently, we discuss the communication and storage costs of the E-Tenon.

Setup. 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 [17], JCE library [19] and Apache Commons IO library [24]. We test modular exponentiation, multiplication and bilinear pairing for 2,000 times and takes the average CPU time in milliseconds.

Table 2. Performance benchmarking of the proposed E-Tenon scheme

Computation Cost of Signing	$\mathcal{T}_{\mathbf{exp}} \approx 2.34\ ms$
Computation Cost of Verification	$\mathcal{T}_{\mathbf{exp}} \approx 2.34\ ms$
Computation Cost of Encryption	$k \cdot \mathcal{T}_{\mathbf{mult}} + 2 \cdot (k + \mathfrak{l}_{MST}) \cdot \mathcal{T}_{\mathbf{exp}} \approx 19.18\ ms$
Computation Cost of Decryption	$(\mathfrak{n}_{MST} + \mathfrak{l}_{AT}) \cdot (2 \cdot \mathcal{T}_{\mathbf{par}} + \mathcal{T}_{\mathbf{exp}} + \mathcal{T}_{\mathbf{mult}})$ $+\mathcal{T}_{\mathbf{mult}} \cdot (2 + m \cdot \mathfrak{n}_{MST}) + \mathcal{T}_{\mathbf{par}} \approx 57.18\ ms$
Communication and Storage Cost of Signature	$2 \cdot \lvert ecc \rvert \approx 320\ bits$
Communication and Storage Cost of Ciphertext	$\{\lvert_{MST}\rvert, 2 \cdot (k + \mathfrak{l}_{MST}) \cdot \lvert\mathbb{G}\rvert\}$
$\mathcal{T}_{\mathbf{exp}}$: 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); $\mathfrak{l}$: number of *external nodes* in the tree; $\mathfrak{n}$: number of *internal nodes* in the tree; $\lvert ecc \rvert$: size of the elliptic curve	

Computation Cost. Table 2 shows the cost for signing, verification, and encryption and decryption. Firstly, the signing and verification algorithms adapted in our model outperform other the relevant algorithms in the state-of-the-art schemes [16,6,26]. Because there is only one exponentiation operation required when an entity signs/verifies the message (the average CPU time for 2000 trials is approximately equal to 2.34 *ms* only). 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 effectively meet this requirement. It is worthy noticeable that the schemes built on the classic CP-ABE 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 $\sum_{l=1}^{c} \mathfrak{l}_{AT_{k_l}} \geq \mathfrak{l}_{MST}$ ($\mathfrak{l}$ denotes the number of attributes/external nodes). 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, that is, instated of encrypting the whole EHR data, we only encrypt a number of constant sized pointers (16 Bytes). This idea reduces the time taken for encryption and decryption considerably, thanks to the use of the *electronic tenon structure*.

Communication and Storage Costs. 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, which means that the size of the ciphertext in E-Tenon system is shorter than other schemes with a series of separate access structures. However, we admit that our protocol requires an extra round of communication during the signing process as compared to other schemes, which is a trade-off for supporting concurrent signing in the multi-user environment as pointed out in MS-BN [4]. That being said, the size of our signature is only $2 \cdot |ecc|$ (note that different schemes may work over a different n-bit elliptic-curve). Following the security discussion in [11], 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. The size of the multiple signature σ is only 320 bits (40 Bytes) in this case. Taken together, the discussion suggests that we have achieved a more secure and reliable protection of EHR without compromising efficiency.

7 Conclusion

In this paper, we 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 as 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 in the direction of protecting 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 independently protect the relationships between EHR blocks with different levels of security, where only legitimate DU with appropriate attributes can decrypt a certain number of pointers and join the open data in a sensible way. 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 result shows that our E-Tenon system does not compromise any security properties while maintaining promising efficiency and flexibility.

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) Advances in Cryptology — EUROCRYPT 2002. pp. 83–107. Springer Berlin Heidelberg, Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_6
2. Bahga, A., Madisetti, V.K.: A cloud-based approach for interoperable electronic health records (EHRs). IEEE Journal of Biomedical and Health Informatics **17**(5), 894–906 (Sep 2013). https://doi.org/10.1109/jbhi.2013.2257818
3. Belguith, S., Kaaniche, N., Laurent, M., Jemai, A., Attia, R.: Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot. Computer Networks **133**, 141–156 (2018). https://doi.org/10.1016/j.comnet.2018.01.036
4. Bellare, M., Neven, G.: 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. p. 390–399. CCS '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1180405.1180453
5. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP '07). pp. 321–334. IEEE (May 2007). https://doi.org/10.1109/SP.2007.11
6. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: International Workshop on Public Key Cryptography. pp. 31–46. Springer (2003). https://doi.org/10.1007/3-540-36288-6_3
7. Camenisch, J., Lehmann, A.: (un)linkable pseudonyms for governmental databases. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. p. 1467–1479. CCS '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2810103.2813658
8. Dagher, G.G., Mohler, J., Milojkovic, M., Marella, P.B.: Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustainable Cities and Society **39**, 283–297 (May 2018). https://doi.org/10.1016/j.scs.2018.02.014
9. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of abe ciphertexts. In: Proceedings of the 20th USENIX Conference on Security (SEC'11). p. 34. USENIX Association, USA (2011). https://doi.org/10.5555/2028067.2028101
10. Kaaniche, N., Laurent, M.: Attribute based encryption for multi-level access control policies. In: SECRYPT 2017: 14th International Conference on Security and Cryptography. vol. 6, pp. 67–78. Scitepress (2017)
11. Koblitz, N., Menezes, A., Vanstone, S.: The state of elliptic curve cryptography. Designs, Codes and Cryptography **19**(2/3), 173–193 (2000)
12. Kumari, A., Kumar, V., Abbasi, M.Y., Kumari, S., Chaudhary, P., Chen, C.M.: CSEF: Cloud-based secure and efficient framework for smart medical system using ECC. IEEE Access **8**, 107838–107852 (2020)
13. Li, J., Chen, X., Li, J., Jia, C., Ma, J., Lou, W.: Fine-grained access control system based on outsourced attribute-based encryption. In: Computer Security – ESORICS 2013. pp. 592–609. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40203-6_33
14. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE transactions on parallel and distributed systems **24**(1), 131–143 (2012). https://doi.org/10.1109/TPDS.2012.97

15. Liu, X., Yi, X.: Privacy-preserving collaborative medical time series analysis based on dynamic time warping. In: Computer Security – ESORICS 2019. pp. 439–460. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-29962-0_21
16. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential aggregate signatures and multisignatures without random oracles. In: Advances in Cryptology - EUROCRYPT 2006, pp. 465–485. Springer Berlin Heidelberg (2006). https://doi.org/10.1007/11761679_28
17. Lynn, B.: Pbc library, https://crypto.stanford.edu/pbc/download.html
18. Maffei, M., Malavolta, G., Reinert, M., Schroder, D.: Privacy and access control for outsourced personal records. In: 2015 IEEE Symposium on Security and Privacy. IEEE (May 2015). https://doi.org/10.1109/SP.2015.28
19. Oracle Technology Network: Java cryptography extension (jce), https://www.oracle.com/java/technologies/javase-jce-all-downloads.html
20. Rezaeibagha, F., Mu, Y.: Distributed clinical data sharing via dynamic access-control policy transformation. International Journal of Medical Informatics **89**, 25–31 (May 2016). https://doi.org/10.1016/j.ijmedinf.2016.02.002
21. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology **4**(3), 161–174 (Jan 1991). https://doi.org/10.1007/bf00196725
22. Shi, J., Lai, J., Li, Y., Deng, R.H., Weng, J.: Authorized keyword search on encrypted data. In: Computer Security - ESORICS 2014, pp. 419–435. Springer International Publishing (2014). https://doi.org/10.1007/978-3-319-11203-9_24
23. Sun, Y., Zhang, R., Wang, X., Gao, K., Liu, L.: A decentralizing attribute-based signature for healthcare blockchain. In: 2018 27th International Conference on Computer Communication and Networks (ICCCN). IEEE (Jul 2018). https://doi.org/10.1109/icccn.2018.8487349
24. The Apache Software Foundation: Apache commons io - uuid generation libraries, https://commons.apache.org/sandbox/commons-id/uuid.html
25. Yang, Y., Liu, X., Deng, R.H.: Lightweight break-glass access control system for healthcare internet-of-things. IEEE Transactions on Industrial Informatics **14**(8), 3610–3617 (Aug 2018). https://doi.org/10.1109/TII.2017.2751640
26. Zhang, Y., Deng, R.H., Han, G., Zheng, D.: Secure smart health with privacy-aware aggregate authentication and access control in internet of things. Journal of Network and Computer Applications **123**, 89–100 (Dec 2018). https://doi.org/10.1016/j.jnca.2018.09.005