2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)

2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS) | 979-8-3503-1293-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00014

PriFR: Privacy-preserving Large-scale File Retrieval System via Blockchain for Encrypted Cloud Data

Hao Ren[1], Guowen Xu[1], Han Qiu[✉,2,3], and Tianwei Zhang[1]
[1]Nanyang Technological University, Singapore
[2]Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China
[3]Zhongguancun Laboratory, Beijing, China
{hao.ren, guowen.xu, tianwei.zhang}@ntu.edu.sg, qiuhan@tsinghua.edu.cn

***Abstract*—As a fundamental and commonly used service, file retrieval has been extensively studied by information retrieval, cryptography, and big data communities. In this paper, we consider the problem of privacy-preserving file retrieval. A new framework named PriFR is proposed by integrating the blockchain and cloud computing infrastructures. The large-scale original files are encrypted and outsourced to the public cloud server. The encrypted retrieval indexes are stored on the full nodes in the blockchain to support traceable and unforgeable retrieval services. This design embraces the benefits brought by both cloud computing and blockchain. For the first time, PriFR decomposes the file retrieval problem into the numerical query and keyword search on the file metadata. In doing so, each file can be characterized more precisely than the traditional keyword search based schemes. In addition to functionality, PriFR only applies fast and lightweight symmetric cryptographical primitives to reach near plaintext retrieval efficiency. In specific, the numerical query is implemented atop order-preserving encryption (OPE) that is perfectly compatible with plaintext indexing techniques. The price for such high efficiency is its vulnerability to inference attacks. Given enough background knowledge, the plaintext can be recovered with high probability. To resist this attack, PriFR injects differential privacy noises into the raw data to offer guaranteed privacy-preserving strength with a negligible extra efficiency cost. The experimental results have demonstrated the effectiveness of PriFR.**

***Index Terms*—Blockchain, big data, private query, searchable encryption, cloud computing.**

I. Introduction

File retrieval [1] is a fundamental and commonly used service and is extensively studied in the areas of information retrieval and big data management. At present, enterprises as well as individual users are generating a vast volume of data at ever-increasing speed. In addition to the growing scale of data volume, data heterogeneity also imposes significant challenges to high-quality data retrieval services. The evidence of this issue is easy to find. For instance, the social networking giants like Facebook and WeChat are gathering images, videos, and texts from users for their profiles or social activities [2]. In this case, heterogeneous data is often stored in a single file. Therefore, the complexity of the file retrieval problem becomes out of the reach of non-experts. A straightforward way is to outsource the files to the public cloud server [1] to enjoy the powerful data processing capability and its large storage space. In addition, the cloud server can provide more reliable computing service than edge devices [3] in the Internet of Things [4]. However, the public cloud server cannot be fully trusted and even manipulates the retrieval results [5]. Thus, we seek to employ blockchain as the file retrieval engine and cloud server as the data warehouse. In doing so, we can not only build a traceable, unforgeable (guaranteed by blockchain) file retrieval system but also offload the heavy storage burden to cloud servers. Such fusion of cloud computing and blockchain infrastructures significantly relieves the tension between efficiency and security [6]–[8].

✉ Corresponding author

However, native cloud computing and blockchain technologies can hardly address privacy concerns [9]. On one hand, the outsourced files may contain sensitive personal information like medical records and social activities [1], [2]. On the other hand, the retrieval users need to keep their retrieval requests and the queried files secret to avoid leakages that may lead to reputation or financial losses [10]. Currently, privacy protection has been not merely a personal preference but a legal requirement. The European general data protection regulation (GDPR) [11] has specified that all the personal data stored on the remote server should be encrypted. To meet these requirements, numerous schemes [5] are proposed to support keyword search [12] over the encrypted files [1]. In specific, an encrypted keyword index is generated and outsourced to the cloud server. The original documents are also encrypted and stored on the cloud. Then, the search user can conduct keywords search on the encrypted files without exposing the selected keywords. This privacy-preserving file retrieval paradigm is well-studied and named searchable encryption (SE) [12], [13]. Despite its high efficiency and provable security, in this paper, we argue that merely considering the keyword search is insufficient for practical file retrieval. For example, a medical record file should definitely contain attributes that are described by numbers including the patient's age, weight, course of disease, etc. In addition, a user may retrieve files that are uploaded within a certain period of time. Unfortunately, these numerical queries can hardly be supported by existing schemes [1], [5], [14] which are indeed needed.

To achieve privacy-preserving and practical file retrieval, in this paper, we propose a new framework dubbed PriFR. In specific, the file retrieval problem is decomposed into the

979-8-3503-1293-5/23/$31.00 ©2023 IEEE
DOI 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00014

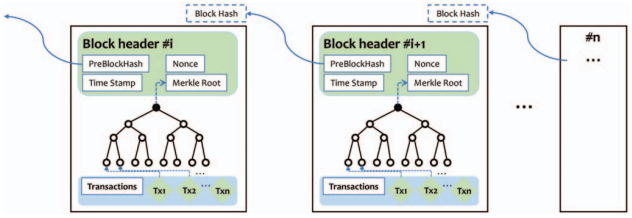


Fig. 1. The schematic diagram of the blockchain.

numerical query and keyword search on the file metadata. Given a file, it is described by several attributes, and all the attributes as a whole are regarded as its metadata. Then, the goal of PriFR is to achieve numerical query and keyword search simultaneously with rich functionality, strong privacy preservation, and high efficiency. In sum, the main technical contributions made by our PriFR are enumerated as follows.

- *Rich functionality.* To the best of our knowledge, PriFR is the first scheme to model the file retrieval problem into metadata search. It can support numerical queries and keyword search simultaneously, which remedies the shortcomings in functionality from previous works [1], [5]. In addition, we use order-preserving encryption (OPE) [15], [16] as the underlying technology for numerical queries. Without loss of generality, PriFR focuses on range query and it can be easily adapted to process other numerical queries including Top-K, kNN, etc.
- *High efficiency.* In PriFR, only fast and lightweight symmetric cryptographical primitives [15], [17] are used to achieve high encryption and decryption efficiency. The proposed OPE-based range query scheme can reach nearly plaintext query efficiency as the plaintext indexing technique is well supported by OPE.
- *Strong privacy preservation.* The high efficiency brought by the OPE-based scheme is not free. Existing inference attacks [15], [16] pose a significant threat to the confidentiality of OPE ciphertexts. To mitigate this problem, PriFR uses differential privacy (DP) noises [18], [19] to further obfuscate the raw data with a provable security guarantee. Meanwhile, the core merit of OPE (i.e., order preservation) is retained.

The remaining contents of this paper are organized as follows. In section II, we give a brief introduction to the technical background of blockchain and the cryptographic primitives. Afterward, the system and threat models are described in section III. The technical details of the numerical range query, keyword search, and the privacy analysis of the proposed scheme are given in section IV. We report the performance in section V. At last, we conclude this paper in section VI.

II. Background and Preliminaries

In this section, we first briefly review the background of blockchain and then describe the cryptographical tools applied in this paper. Note that, due to the limitation of space, not every detail of the related techniques is involved. In specific, the main characters of each preliminary and their role in the proposed PriFR are illustrated.

A. Blockchain

Blockchain is initiated by Nakamoto [20] to build the first peer-to-peer cryptocurrency system bitcoin. It is an append-only distributed digital hyperledger, that eliminates the trusted party while supporting undeniable and traceable transactions for individual participants in the network. As shown in Fig. 1, the blockchain is constructed by data blocks and is secured by cryptographical techniques. All the participants in the network run a consensus protocol to decide who generates the new block. At present, two mainstream consensus protocols are widely used, which are proof of work (PoW) [20] and proof of stake (PoS) [21]. In PoW based blockchain, the participant who is the first to solve a cryptographical puzzle will be authorized to generate a new block. Each block comprises two types of data. One is the block header and the other is transaction data. The block header comprises the following four items.

- **PreBlockHash.** The is the hashing value of the whole previous block used to form the chain. Once a block is tempered, any node in the network can detect such behavior by simply computing and checking the block hash values. Thus, the block integrity is preserved.
- **Nonce.** It is generated by miners in the PoW blockchain, which can be regarded as a solution for PoW puzzle.
- **Time Stamp.** It records the time when the corresponding block is appended.
- **Merkle Root.** Each transaction in the same block is hashed as a leaf node of a Merkle hash tree [22]. To be succinct, the block header only includes the root. This item allows participants in the network to verify the integrity of the transactions.

The transaction data is stored in the block body. It records the payer and payee's addresses, transaction value, and a signature of the transaction generated with the payer's private key. Beyond transferring cryptocurrency, the current *smart contracts* [7] enables versatile self-executing agreements that are written directly into lines of code. In theory, once the functions of the submitted smart contracts are well-defined, any algorithm or protocol can be instantiated [10]. Meanwhile, smart contracts naturally inherit the benefits brought by blockchain, including integrity preservation and tamper resistance. In this paper, we mainly investigate the privacy-preserving file retrieval problem on the encrypted data using smart contract enabled blockchain [1], [23], [24]. In specific, the blockchain and smart contract serve as the supporting platform for our proposed PriFR.

B. Cryptographical Primitives

To preserve the privacy of the outsourced cloud data while offering efficient and functionality-rich file retrieval services, we adopt the following cryptographical tools as the building blocks. Note that, in this section, we do not traverse every primitive used in PriFR. The corresponding remaining primitives will be introduced in the related sections.

Symmetric Searchable Encryption (SSE). SSE is first proposed by Song et al. [13] to enable privacy-preserving keyword search over the encrypted files. SSE [5] has been intensively studied in the past two decades. A typical SSE system [5] mainly comprises two entities, which are the data owner and remote server (i.e., cloud server in articles [5], [8]). We depict the basic algorithms for SSE as follows.

- *System initialization.* The data owner first generates private keys for encryption. Then, it generates an inverted index for the keyword dictionary. For instance, if a file contains a specific keyword, its identifier (ID) will be appended to the corresponding keyword. Afterward, the inverted index is encrypted using a customized cryptosystem [5]. The files are encrypted using standard symmetric encryption (e.g., AES). The encrypted index along with the encrypted files is outsourced to the remote server.
- *Trapdoor generation.* Data owner uses private keys to generate a trapdoor for the selected keywords. Then, it uploads the trapdoor to the remote server.
- *Search on the encrypted files.* Upon receiving the trapdoor, the server executes the predefined protocol to conduct keyword matching on the encrypted index and trapdoor. If any keyword is matched, the corresponding encrypted files will be returned to the data owner. Then, the data owner simply decrypts them as the search result.

Order-preserving Encryption (OPE). The ciphertexts generated by OPE [15] can preserve the order information. For example, given two messages $m_1 > m_2$ and their OPE generated ciphertexts c_1, c_2, we have $c_1 > c_2$. This property is appealing and motivates its application for privacy-preserving queries [15], [16]. In this paper, we use the triple $\{\mathsf{OPE.Key}, \mathsf{OPE.Enc}, \mathsf{OPE.Dec}\}$ to indicate the secret key generation, encryption, and decryption algorithms for OPE. The technical details can be found in [15], [16].

Differential Privacy (DP). DP is regarded as the de-facto standard privacy-preserving mechanism that offers a mathematical quantifiable measure of the protection strength [18]. Intuitively, the user can tune the parameter named privacy budget to seek a satisfactory trade-off between privacy protection and data utility [18]. Currently, there are mainly two trendy DP paradigms, central (CDP) and local (LDP) [19]. The primary difference between CDP and LDP is the trust setting. In CDP, a trusted aggregator is needed to perturb the collected private data from the individuals. While LDP allows individuals directly add the noises to the raw data. The succinct definitions of CDP and LDP are given as follows, respectively.

Definition 2.1 (Central Differential Privacy, CDP): Given a probabilistic algorithm $\mathcal{A} : \mathcal{U}^n \to \mathcal{V}$, for all adjacent datasets $D, D' \in \mathcal{U}^n$, all subranges of the output $\mathcal{S} \in \mathcal{U}$, $\mathcal{A}$ is ϵ-DP if the following inequation holds.

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in \mathcal{S}]. \tag{1}$$

In the context of DP, the notion of two adjacent datasets mean that D and D' only differ in one entry.

Definition 2.2 (Local Differential Privacy, LDP): Given a probabilistic algorithm $\mathcal{A} : \mathcal{U} \to \mathcal{V}$, for any pair of private inputs $d, d' \in \mathcal{U}$, all subranges of the output $\mathcal{S} \in \mathcal{U}$, $\mathcal{A}$ is ϵ-LDP if the following inequation holds.

$$\Pr[\mathcal{A}(d) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{A}(d') \in \mathcal{S}]. \tag{2}$$

Note that, the above Definitions 2.1, 2.2 are the standard DP model. In this paper, we relax the DP model by incorporating the data distances to strengthen the security of OPE [16].

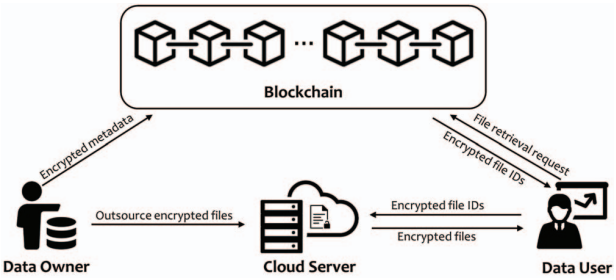


Fig. 2. System model.

III. Problem Statements

In this section, we first introduce the system model and elaborate on the roles of each entity to sketch the workflow for PriRF. Afterward, the formal threat model is given.

A. System Model

As shown in Fig. 2, PriFR comprises four entities. Below, we elaborate on the role of each entity and briefly review the workflow.

- **Data Owner (DO).** It is the owner of the files that outsources the file storage and retrieval services to the public cloud and blockchain. **DO** first generates metadata for each file and encrypts them as the retrieval index. Then the encrypted metadata will be uploaded to the blockchain (full node). The original files along with their identifiers (IDs) are encrypted using fast symmetric encryption (e.g., AES) and outsourced to the cloud server.
- **Data User (DU).** It is authorized by **DO** that can launch file retrieval requests to the blockchain. **DU** receives the encrypted IDs. Then, the encrypted IDs are sent to the cloud server. At last, the corresponding encrypted files are returned to **DU** as the final result.
- **Blockchain (BC).** It stores the encrypted metadata and ID for each file. Once a retrieval request arrived, **BC** conducts secure metadata matching by invoking pre-set smart contracts and returns the encrypted IDs to **DU**.
- **Cloud Server (CS).** It plays the role of a public data warehouse. **CS** stores the large-scale encrypted files and their encrypted IDs. Upon receiving the retrieved file IDs from **DU**, **CS** returns the corresponding encrypted files.

B. Threat Model

In this paper, we consider **BC** and **CS** to be semi-honest (i.e., honest but curious), which is practical for real-world scenarios [8], [23]–[25]. In specific, the **BC** and **CS** will follow the pre-set protocols strictly without any deviation. However,

they may conduct passive inference on the collected private data to mine some desired information, for economic interests, or for voyeurism [8]. **DO** is the owner of all the data and thus is assumed to be fully trusted. **DU** is the authorized user of the data and is also considered to be fully trusted. Note that, in works [5], **DU** is considered to be semi-honest or malicious. Therefore, access control [26] and verification mechanisms [24] are needed, which is out of the scope of this paper.

IV. Proposed Scheme

In this section, we present the technical details of PriFR. As discussed in Section I, the problem of private file retrieval is modeled as metadata matching. This problem is then decomposed into three sub-problems, which are private numerical range query, encrypted keyword search, and final result revealing. We elaborate on each subroutine in the following subsections.

A. Private Range Query

There are two mainstream technical routes [27] on solving private numerical range queries. The first route is comparing the queried range bounds with each data record. For instance, given a query range $[a, b], a, b \in \mathbb{R}$ and database D with data records $x_i \in \mathbb{R}, i \in |D|$, the results of a query is actually directly computed by inspecting inequation $a \leq x_i \leq b$ without accessing the original values of a, b, x_i. To support the numerical comparison over the ciphertext domain, OPE [15] as a lightweight crypto-system, is extensively used. The second route is converting the range query problem into a set membership test. For instance, the query range $[1, 10]$ can be represented by a set of numbers like $\{1, 1.1, 1.2, ..., 10\}$. Then, we can determine the query result by inspecting whether x_i belongs to the given set. To implement such a set membership test, exiting encrypted keyword search (e.g., SSE) efforts [27] can be exploited if we treat x_i and the elements in set $\{1, 1.1, 1.2, ..., 10\}$ as keywords.

Why we choose OPE as the cornerstone. It is straightforward if we turn to use existing SSE methods for answering encrypted range queries. However, the high efficiency and scalability can hardly be preserved. First, the current SSE method [5], [27] works well for keyword search, but when the query range becomes large, the size of queried keywords expands at least linearly which ultimately leads to significant additional costs. Second, the efficient query indexing technique like B+ tree can hardly be compatible with SSE implementation. Therefore, to conquer these two issues, we use OPE as the building block for PriFR. Compare to the SSE method [27], due to the use of OPE, PriFR can not only reduce the encryption and communication costs but also perfectly support any plaintext query boosting technique.

Enhancing the security of OPE by DP. As the OPE ciphertexts preserve the order information, inference attacks [15] have long been the main threat, that can even recover partial plaintexts. Introducing DP noises [16] can offer a quantifiable guarantee against inference attacks and high utility simultaneously. The key idea is encoding the raw data using relaxed DP mechanisms, then invoking OPE to encrypt the encoded data. In the following paragraphs, we first give the definitions of relaxed DP mechanisms. Based on the relaxed DP, a new raw data encoding method is introduced [16]. At last, we give the detail of the private range query protocol.

Definition 4.1 (Distance-based Central Differential Privacy, dCDP): Given a probabilistic algorithm $\mathcal{A} : \mathcal{U}^n \to \mathcal{V}$, for all adjacent datasets $D, D' \in \mathcal{U}^n$ that differ in one element d_i, d_i', respectively; all subranges of the output $\mathcal{S} \in \mathcal{U}$, $\mathcal{A}$ is ϵ-dCDP if the following inequation holds.

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\epsilon |d_i - d_i'|} \Pr[\mathcal{A}(D') \in \mathcal{S}]. \tag{3}$$

Definition 4.2 (Distance-based Local Differential Privacy, dLDP): Given a probabilistic algorithm $\mathcal{A} : \mathcal{U} \to \mathcal{V}$, for all pair of private values d, d', all subranges of the output $\mathcal{S} \in \mathcal{U}$, $\mathcal{A}$ is ϵ-dLDP if the following inequation holds.

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\epsilon |d - d'|} \Pr[\mathcal{A}(D') \in \mathcal{S}]. \tag{4}$$

Definition 4.1 and 4.2 are distance-based relaxed DP models. Intuitively, less noise will be injected into the data if the neighbor databases or pair of values have a short distance. Thus, ϵ-dCDP and ϵ-dLDP can make a subtle balance between the data utility and privacy protection for OPE. Below, based on the ϵ-dLDP model, we show the technical details of the raw data encoding method, dubbed as $\mathsf{Cd}_{\mathsf{dLDP}}$.

Construction of $\mathsf{Cd}_{\mathsf{dLDP}}$. Assume $\mathcal{I}$ is the input distribution, and its prior distribution is $\mathcal{D}$. The output domain $\{od_1, ..., od_t\}$ is denoted as $\mathcal{O}$. Let m be the number that needs to be encoded and ϵ be the privacy budget. $\mathcal{P}$ be a t-partition $\{[u_1, v_1], ..., (u_t, v_t]\}$ on $\mathcal{I}$. The output encoding is written as OE. $\mathsf{Cd}_{\mathsf{dLDP}}$ is constructed by following two steps.

Step 1. For all $i \in [t]$, computes the weighted median for each interval $(u_i, v_i]$ according to $\mathcal{D}$. If $\mathcal{D}$ is not available, the uniform distribution will be adopted.

Step 2. For all $m \in \mathcal{I}$, and $i \in [t]$, computes the encoding probability distribution for each m as:

$$f_{m,i} = \Pr[\mathsf{Cd}_{\mathsf{dLDP}}(m, \mathcal{P}, \epsilon)] = \frac{e^{-|m - w_i| \cdot \epsilon/2}}{\sum_{j=1}^{t} e^{-|m - w_j| \cdot \epsilon/2}}. \tag{5}$$

At last, the output encoding is randomly sampled from $f_m = \{f_{m,1}, ..., f_{m,t}\}$ as $\mathrm{OE} \leftarrow f_m$. As shown in Equation 5, the classic exponential DP mechanism [18] is used.

By integrating $\mathsf{Cd}_{\mathsf{dLDP}}$, the DP-enabled OPE scheme can be implemented with the assistance of authenticated encryption [17]. To be succinct, we only give the triple of authenticated encryption $\{\mathsf{AE.Key}, \mathsf{AE.Enc}, \mathsf{AE.Dec}\}$ to represent the key generation, encryption, and decryption algorithms, respectively. We write the augmented OPE as $\mathsf{OPE}\epsilon$. The syntax of $\mathsf{OPE}\epsilon$ is defined as follows.

- $\mathsf{OPE}\epsilon.\mathsf{Key}(\lambda)$. It invokes $\mathsf{SK}_{\mathsf{OPE}} \leftarrow \mathsf{OPE.Key}(\lambda)$ and $\mathsf{SK}_{\mathsf{AE}} \leftarrow \mathsf{AE.Key}(\lambda)$, where λ is the security parameter. The private keys $\{\mathsf{SK}_{\mathsf{OPE}}, \mathsf{SK}_{\mathsf{AE}}\}$ are returned.
- $\mathsf{OPE}\epsilon.\mathsf{Enc}(\cdot)$. It encrypts the data $m \in \mathcal{I}$ by computing $\overline{\mathrm{OE}} \leftarrow \mathsf{Cd}_{\mathsf{dLDP}}(m, \mathcal{P}, \epsilon/2)$, $(\mathsf{SK}_{\mathsf{OPE}}{}', c_0) \leftarrow$

$\mathsf{OPE.Enc}(\mathsf{SK}_{\mathsf{OPE}}, \overline{\mathrm{OE}}, \Pi)$, $c_1 \leftarrow \mathsf{AE.Enc}(\mathsf{SK}_{\mathsf{AE}}, m)$. The returned ciphertext is a triple $\mathsf{CT} = (\mathsf{SK}_{\mathsf{OPE}}{}', c_0, c_1)$.

- $\mathsf{OPE}\epsilon.\mathsf{Dec}(\cdot)$. It takes the private keys and ciphertext triple as the input and then computes $m \leftarrow \mathsf{AE.Dec}(\mathsf{SK}_{\mathsf{AE}}, c_1)$, $\overline{\mathrm{OE}} \leftarrow \mathsf{OPE.Dec}(\mathsf{SK}_{\mathsf{OPE}}, c_0)$ to obtain the plaintext pair $(m, \overline{\mathrm{OE}})$.

Range query using $\mathsf{OPE}\epsilon$. In this part, we illustrate the implementation details for the range query protocol built atop $\mathsf{OPE}\epsilon$. Recall that, in the context of OPE, each ciphertext is unique. In addition, $\mathsf{OPE}\epsilon$ has introduced randomness into the ciphertexts. Thus, a query issuer needs to maintain the state information for each plaintext. For a simple example, given the plaintext set $M = \{m_1, ..., m_t\}$, its corresponding plaintext set $N = \{n_1, ..., n_t\}$, then a query issuer needs to keep the maximum and minimum ciphertexts in N for each m_i. They are written as $\max \langle m_i \rangle$ and $\min \langle m_i \rangle$. In the remainder of this paper, we use $\langle m \rangle$ to denote the ciphertext of data m. Thus, to issue a range query $[a, b]$, the queried records should belong to $[\min \langle a \rangle, \max \langle b \rangle]$. This principle remains unchanged when $\mathsf{OPE}\epsilon$ is applied. Below, we depict the detailed design of the range query protocol.

We define the notations and the input/output items. The dataset with t records is written as M. In M, each record is denoted as $(m_i, k_i), i \in [t]$. m_i is the private data that needs to be encrypted, and k_i is the other associated data. The query issuer can decide whether to encrypt k_i or not. The private key pair of $\mathsf{OPE}\epsilon$ is $\{\mathsf{SK}_{\mathsf{OPE}}, \mathsf{SK}_{\mathsf{AE}}\}$. $\mathcal{P}$ represents the partition used for $\mathsf{OPE}\epsilon$. Assume the input is a range query $[a, b]; a, b \in \mathcal{I}$, the output is a set of queried records $R = \{k_i | m_i \in [a, b], (m_i, k_i) \in M\}$. The range query can be achieved by the following three phases.

Phase 1. Initialization : 1). Data owner (**DO**) first prepares the input dataset $M = \{m_1, ..., m_t\}$. Then it invokes $N \leftarrow \mathsf{OPE}\epsilon(\mathsf{SK}_{\mathsf{OPE}}, \mathsf{SK}_{\mathsf{AE}}, M, \Pi, \mathcal{P}, \epsilon/2)$. The ciphertext set is then sent to the service provider (i.e., **BC**). 2). For all output encoding $\mathrm{OE} \in \mathcal{O}$, the aforementioned *state information* is generated by **DO** and shared with **DU**.

Phase 2. Issue and answer range query : 1). **DU** generate an encrypted query as $Q \leftarrow [\min \langle \mathcal{P}(a) \rangle, \max \langle \mathcal{P}(b) \rangle]$ according to the state information. Then, **DU** sends it to **BC**. 2). Upon receiving the query, **BC** simply compares the values between the encrypted dataset N and Q. The returned query result can be written as $R' \leftarrow \{(x_{i0}, x_{i1}) | i \in [t]; x_{i0} \in [\min \langle \mathcal{P}(a) \rangle, \max \langle \mathcal{P}(b) \rangle]\}$. At last, R' along with the corresponding encrypted file IDs are sent back to **DU**. Note that, this step can be implemented using any plaintext indexing technique like B+ tree to boost query efficiency.

Phase 3. Result filtering : As the DP noises are injected into the dataset, the query result inevitably may contain a few false positive items. Thus, **DU** should filter out these wrong items. If $R' \neq \phi$, for all $x_{i1} \in R'$, **DU** first invokes $x'_i \leftarrow \mathsf{AE.Dec}(\mathsf{SK}_{\mathsf{AE}}, x_{i1})$. Then if $x'_i \in [a, b]$, the corresponding file ID will be accepted as a correct item. Otherwise, it will be asserted as a false item and abandoned. At last, **DU** uploads the correct file IDs to **CS** and fetches the encrypted original files. The files will be decrypted as the final result.

Note that, the above private range query protocol can be implemented by using smart contracts. In specific, once the encrypted dataset is prepared, **DO** may generate a smart contract to request query service to **BC**. Then the integrity of the protocol is guaranteed.

Remark. In sum, the encoding method $\mathsf{Cd}_{\mathsf{dLDP}}$ is used as the subroutine of $\mathsf{OPE}\epsilon.\mathsf{Key}(\lambda)$, and $\mathsf{OPE}\epsilon.\mathsf{Key}(\lambda)$ is applied to encrypt the data records and to support private range query. To ease the understanding, we choose to illustrate the entire protocol in a bottom-up manner rather than summarizing them into a single algorithm.

B. Encrypted Keyword Search

It is insufficient in terms of functionality if a file retrieval system only supports numerical range queries. Thus, we seek to use the SSE method [5], [27] to offer keyword search service as a complement to PriFR. In this paper, we do not aim to present a novel SSE scheme, which has been extensively studied in the past two decades. Therefore, we propose to use the existing SSE scheme [27] and integrate it into our blockchain-based PriFR. The pipeline of keyword searches for PriFR roughly contains the following four steps. 1) **DO** extracts keywords from the files and builds a plaintext dictionary. Then it constructs an encrypted inverted index Ind atop the dictionary. The encrypted files along with the file IDs are uploaded to **CS**. Ind is sent to **BC**. The private key is shared with **DU**. 2). When **DU** needs to issue a keyword search, it uses the private key to generate a trapdoor Trap for selected keywords. Afterward, Trap is sent to **BC** for searching on Ind. 3). The search result (encrypted file IDs) will be returned to **DU**. 4). At last, **DU** uploads the encrypted file IDs to **CS** and fetches the encrypted original files. The files will be decrypted as the final result. The encrypted keyword search service can also be implemented by smart contracts [7].

C. Final Result Revealing

In Subsection IV-A and IV-B, the subroutines for numerical range query and keyword search are introduced, respectively. However, solely using one of the functions can hardly support metadata matching for PriFR. Thus, when both range query and keyword search are needed, **BC** should hold the intermediate results and filter out the matched encrypted file IDs. It is trivial if these two types of queries are conducted by the same node in **BC**, a simple set intersection operation can reveal the result. If these two services are provided by two distinct nodes, we need to handle this problem carefully for different privacy requirements. In case the intermediate results (i.e., encrypted file IDs) are considered non-sensitive information in **BC**, one party can just share the IDs with another party. Then, the result can be revealed by a set intersection operation. However, in some scenarios, this information is considered sensitive. In another word, the set intersection needs to be revealed without leaking the remaining elements. This problem can be solved by invoking a private set intersection (PSI) [28] protocol. It offers privacy-enhancing property at the cost of significant additional computational/communication costs. This is out of the scope

of this paper, the interested readers are referred to [28]. Note that, as the last step, **DU** still needs to filter out the false positives incurred by $\mathsf{OPE}\epsilon$.

D. Privacy Analysis

PriFR is built atop the proven secure cryptographical primitives as discussed in Section IV. Therefore, the privacy of the original file data as well as the metadata is strictly preserved, especially in the DP-enabled range query protocol. Below, we give a brief discussion on the privacy issue for all the shared data in the system.

- *Privacy of* $\mathsf{OPE}\epsilon$ *encrypted data for the range query.* The data encrypted by $\mathsf{OPE}\epsilon$ can generates indistinguishable ciphertexts under frequency-analyzing ϵ-dCDP ordered choose plaintext attacks (ϵ-IND-FA-OCPA) [16]. If the privacy budget ϵ is given, the attack resistance strength can be rigorously guaranteed (ϵ-dCDP). Thus, the privacy of $\mathsf{OPE}\epsilon$ encrypted data is well preserved.
- *Privacy of the SSE encrypted data.* At present, the latest SSE scheme can not only provide provable privacy protection in the static setting but also offers forward and backward privacy even when **DO** can dynamically update the encrypted files [5], [27]. In addition, if a novel attack can break through the existing scheme, PriFR can just use an alternative secure scheme.
- *Privacy of the outsourced file data.* The files stored on the **CS** are encrypted using advanced encryption standard (AES), which has been commonly used and proven to be secure if the length of the private key is set appropriately.

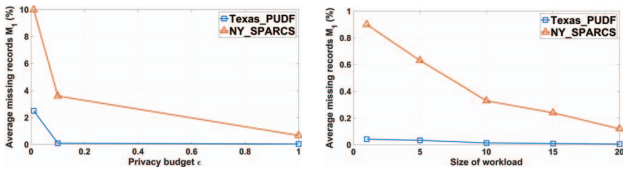
V. Performance Evaluation

In this section, we elaborate on the performance of the proposed PriFR. Due to the limitation of space, in this conference version, we mainly report the impacts of differential privacy noises on query accuracy for $\mathsf{OPE}\epsilon$ and the extra records processed. In specific, we first give the details of experimental settings and the used datasets. Then, the effects of the privacy budget and workload are reported, respectively. At last, we give some additional discussions on the other issues related to the performance of PriFR.

A. Experimental Settings and Datasets

We implement the experiments on the local computing matching (Server) with Intel (R) Xeon(R) E5-2697 v3 2.6GHz CPUs with 28 threads on 14 cores and 64GB memory. The programming language is Python and the open-source library pyca/cryptography is applied. Below, we give a brief introduction to the testing data about their features.

- Texas_PUDF [29]. It is a public medical dataset released by the state of Texas. PUDF stands for public use data file. Texas_PUDF contains large-scale hospital discharge data in Texas collected from 1999. In specific, we use the attribute PAT_ZIP that is zipcode within domain $[70601, 88415]$. It roughly contains 730K records.
- NY_SPARCS [30]. It is also a public medical dataset released by the state of New York. SPARCS stands for statewide planning and research cooperative system. It has roughly 2500K hospital inpatient discharge records. The attribute length_of_stay ranging in $[1, 120]$ is used.



(a) Impact of ϵ on M_1 (b) Impact of workload on M_1

Fig. 3. Performance evaluated by metric M_1 (%).

Tow mainstream metrics are adopted [16], that are the relative proportion of missing records M_1 and the proportion of additional records M_2 processed by $\mathsf{OPE}\epsilon$. Specifically, they are defined as follows.

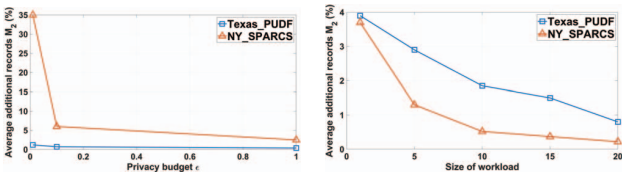
$$\mathsf{M}_1 = \frac{\text{missing records}}{\text{correct records}}\%;\ \mathsf{M}_2 = \frac{\text{additional records}}{\text{size of dataset}}\%. \quad (6)$$

The first metric M_1 can capture the false negatives. For instance, the number 9.7 should be included if the query range is $[1, 10]$. However, if the added DP noise is larger than 0.3, 9.7 will be excluded falsely. Note that, the false positives can be easily filtered out by **DU**. The second metric M_2 indicates the processed extra records. Since the underlying encryption techniques including OPE and AE used in $\mathsf{OPE}\epsilon$ are the same as the previous scheme [15]. Thus, the scale of additional computational/communication overheads brought by $\mathsf{OPE}\epsilon$ can be characterized by this metric.

B. Experimental Results

Recall that the standard OPE [15] can perfectly preserve the order information for all the ciphertexts. Thus, the mainstream plaintext indexing method for range query including R-tree, B+ tree, etc., can be directly constructed over the OPE ciphertext domain. Thus, OPE is regarded as the fastest cryptographic primitive for private numerical queries. In PrivFR, the DP noises are added to obtain ϵ-dCDP privacy guarantee. As the order information is mildly obfuscated by the added noises, two factors are incurred. First, some false negatives (missing correct records) are inevitable when the noisy plaintexts jump out of the original range. This factor is characterized by the metric M_1. Second, due to the use of range partition algorithm $\mathcal{P}$ and a few records may be noisily mapped to the intervals. Thus, a few additional records need to be processed for issued queries. As a result, a mild additional performance cost is imposed. We use the metric M_2 to evaluate this factor.

Performance evaluated by metric M_1. We first report the impact of privacy budget ϵ on M_1. In theory, with the decrease of ϵ, the noise scale will be compressed which ultimately leads to fewer missing records. The results reported in Fig. 3(a) are consistent with this principle. For Texas_PUDF, when $\epsilon = 1$, M_1 is roughly 0.033%. It indicates that on average only around 3.3 records may be missed in 10^4 correct records. In addition, $\epsilon = 1$ is commonly chosen in real



(a) Impact of ϵ on M_2 (b) Impact of workload on M_2

Fig. 4. Performance evaluated by metric M_2 (%).

scenarios [18]. For NY_SPARCS, M_1 will be lower because the generated intervals (i.e,. $\mathcal{P}$) are smaller which leads to more missing records. In specific, M_1 is roughly 0.67% when $\epsilon = 1$. The trend of M_1 for NY_SPARCS is the same as Texas_PUDF. Another way to reduce M_1 is to issue a batch of queries at once. Thus, increasing the workload can improve the utility of PriFR. As shown in Fig. 3(b), M_1 drops from 0.042% to 0.006% and 0.9% to 0.12% for Texas_PUDF and NY_SPARCS, respectively when the workload varies from 1 to 20. Thus, PriFR fits more to the large workload.

Performance evaluated by metric M_2. For M_2, the trends are similar to M_1 when varying either parameter ϵ or the workload, which is demonstrated in Fig. 4(a) and Fig. 4(b). Specifically, for Texas_PUDF, the size of additional records retrieved by PriFR is mild (roughly 1.2%) when $\epsilon = 0.01$. If ϵ is set to 1, M_1 drops to around 0.39%. For the same ϵ, for NY_SPARCS, M_2 is roughly 2.5%. It means that on average nearly 2.5 additional records are retrieved in 100 records. Thus, the extra time cost is negligible. Similarly, as shown in Fig. 4(b), with the increasing of workload, the reduction of M_2 is significant for both Texas_PUDF and NY_SPARCS. It drops from 3.9% to 0.8% and 3.7% to 0.22% for Texas_PUDF and NY_SPARCS when varying the workload from 1 to 20, respectively. Note that, for Texas_PUDF, M_2 will be lower because the generated intervals (i.e,. $\mathcal{P}$) are larger which leads to fewer missing records. Due to this reason, the curve of Texas_PUDF changes more moderately than NY_SPARCS, which is demonstrated in Fig. 4.

Computational complexity. In this part, we discuss the computation complexity of the proposed OPE-based numerical query scheme. Assume that, the number of data records and the number of partitions are n, t, respectively. At a high level, the data domain is partitioned into t intervals. Then, data records are mapped into intervals. To conceal the original data and to enable decryption, each record is encrypted by a symmetric encryption method [17]. Obviously, the computational cost of generating an OPE generated database is linear to n. Assume that the time cost of domain partition (i.e., OPE encoding) is T_P, and the encryption cost of one record is T_E. Then, the computation complexity is $T_P + nT_E$, where $T_P \sim O(t)$. In this paper, the differential privacy noises are introduced into the domain partition algorithm, which brings negligible cost. Moreover, PriFR only needs to invoke the partition algorithm once for a given data domain. As the numerical range query can be directly conducted over the OEP encrypted data, no additional computational cost is incurred [16]. Thus, in theory, and practice, OPE is regarded as the most efficient cryptosystem for privacy-preserving numerical range queries.

Remark. SSE [5] is extensively studied in the past decades, and its theoretical best-possible efficiency/security trade-offs are provided. The goal of this paper is not to propose a new SSE scheme. Interested readers may refer to [1], [5], [27] for the performance of SSE.

Discussion. As the data domain partition and record mapping algorithm $\mathsf{Cd}_{\mathsf{dLDP}}$ introduces differential private noises, some of the data records may be mapped into the wrong partition. Thus, false positives are inevitable. Fortunately, the returned encrypted results should be decrypted by using the private key $\mathsf{SK}_{\mathsf{AE}}$. Thus, this issue can be easily addressed by filtering out the wrong results over the plaintext domain by **DU**. There should be no technical challenge. In theory, the false positive ratio can be calibrated by the differential private [18] noises. For instance, the less ϵ the larger the false positive ratio is. This type of error is captured by metric M_2.

VI. Conclusions and Future Works

In this paper, we proposed a practical and privacy-preserving file retrieval system PriFR by integrating cloud computing and blockchain. PriFR cannot only enjoy the rich storage resources of the cloud server but also leverage the blockchain infrastructure for secure and reliable file metadata retrieval. Rich functionalities are achieved with strong privacy preservation at mild extra costs. The new framework and pipeline proposed in PriFR may motivate future research on secure big data retrieval with the assistance of blockchain. In the future, we will investigate blockchain-enabled big data processing.

Acknowledgment

The authors would thank the anonymous reviewers. This work has been supported in part by the Algorand Centres of Excellence programme managed by Algorand Foundation, Singapore Ministry of Education (MOE) AcRF Tier 2 MOE-T2EP20121- 0006 and NTU Start-up grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Algorand Foundation.

References

[1] K. Zhang, X. Wang, J. Ning, M. Wen, and R. Lu, "Multi-client boolean file retrieval with adaptable authorization switching for secure cloud search services," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[2] T. Gao and F. Li, "Machine learning-based online social network privacy preservation," in *Proceedings of the ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2022, pp. 467–478.

[3] H. Qiu, M. Qiu, and R. Lu, "Secure v2x communication network based on intelligent pki and edge computing," *IEEE Network*, vol. 34, no. 2, pp. 172–178, 2019.

[4] H. Qiu, Q. Zheng, G. Memmi, J. Lu, M. Qiu, and B. Thuraisingham, "Deep residual learning-based enhanced jpeg compression in the Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2124–2133, 2020.

[5] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, pp. 1–51, 2014.

[6] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.

[7] W. Zou, D. Lo, P. S. Kochhar, X.-B. D. Le, X. Xia, Y. Feng, Z. Chen, and B. Xu, "Smart contract development: Challenges and opportunities," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2084–2106, 2019.

[8] A. Yang, J. Xu, J. Weng, J. Zhou, and D. S. Wong, "Lightweight and privacy-preserving delegatable proofs of storage with data dynamics in cloud storage," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 212–225, 2021.

[9] H. Ren, H. Li, D. Liu, G. Xu, N. Cheng, and X. Shen, "Privacy-preserving efficient verifiable deep packet inspection for cloud-assisted middlebox," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1052–1064, 2022.

[10] S. Steffen, B. Bichsel, R. Baumgartner, and M. Vechev, "Zeestar: Private smart contracts by homomorphic encryption and zero-knowledge proofs," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2022, pp. 179–197.

[11] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "GDPR-compliant personal data management: A blockchain-based solution," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1746–1761, 2019.

[12] C. Huang, D. Liu, A. Yang, R. Lu, and X. Shen, "Multi-client secure and efficient dpf-based keyword search for cloud storage," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[13] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2000, pp. 44–55.

[14] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 870–885, 2018.

[15] F. Kerschbaum, "Frequency-hiding order-preserving encryption," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 656–667.

[16] A. Roy Chowdhury, B. Ding, S. Jha, W. Liu, and J. Zhou, "Strengthening order preserving encryption with differential privacy," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022, pp. 2519–2533.

[17] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1. 2: Lightweight authenticated encryption and hashing," *Journal of Cryptology*, vol. 34, pp. 1–42, 2021.

[18] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[19] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy in blockchain technology: A futuristic approach," *Journal of Parallel and Distributed Computing*, vol. 145, pp. 50–74, 2020.

[20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[21] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Proceedings of the Annual International Cryptology Conference (CRYPTO)*. Springer, 2017, pp. 357–388.

[22] R. C. Merkle, "Protocols for public key cryptosystems," in *Secure Communications and Asymmetric Cryptosystems*. Routledge, 2019, pp. 73–104.

[23] C. Xu, C. Zhang, and J. Xu, "vchain: Enabling verifiable boolean range queries over blockchain databases," in *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2019, pp. 141–158.

[24] H. Wang, C. Xu, C. Zhang, J. Xu, Z. Peng, and J. Pei, "vchain+: Optimizing verifiable blockchain boolean range queries," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 1927–1940.

[25] H. Ren, H. Li, D. Liu, G. Xu, and X. S. Shen, "Enabling secure and versatile packet inspection with probable cause privacy for outsourced middlebox," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2580–2594, 2022.

[26] D. Han, Y. Zhu, D. Li, W. Liang, A. Souri, and K.-C. Li, "A blockchain-based auditable access control system for private data in service-centric IoT environments," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3530–3540, 2022.

[27] C. Zuo, S.-F. Sun, J. K. Liu, J. Shao, J. Pieprzyk, and L. Xu, "Forward and backward private DSSE for range queries," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 328–338, 2022.

[28] M. Chase and P. Miao, "Private set intersection in the internet setting from lightweight oblivious PRF," in *Proceedings of the Annual International Cryptology Conference (CRYPTO)*. Springer, 2020, pp. 34–63.

[29] Texas state, "Hospital discharge data public use data file," http://www.dshs.state.tx.us/THCIC/Hospitals/Download.shtm/, 2013.

[30] New York state, "Hospital inpatient discharges." https://health.data.ny.gov/Health/HospitalInpatient-Discharges-SPARCS-De-Identified/u4ud-w55t/, 2012.