Towards Privacy-Preserving Forensic Analysis for Time-series Medical Data

Xiaoning Liu, Xingliang Yuan, and Joseph Liu
Faculty of Information Technology, Monash University, Australia
maggie.liu@monash.edu, xingliang.yuan@monash.edu, joseph.liu@monash.edu.

***Abstract*—Electronic medical record (EMR) forensics is at the forefront of both academia and industry, and has dominated increasingly important role in the fast revolutionized digital forensics area. Upon the severe financial loss and user privacy revealing caused by data breaches, protecting the forensic medical records only being mined by authorized investigators and data confidentiality is deemed essential. Standard encryption technique can ensure the end-to-end data security, yet restricting the functionality in forensic analyzing. How to proceed similarity match over forensic physiological data in a private manner is intrinsically challenging, because the natural properties of such medical data are high-dimensional and times series related. In this paper, we propose a secure framework to proceed similarity match over encrypted physiological time-series data. Our framework resorts to an advanced similarity search algorithm, aka stratified locality-sensitive hashing (SLSH) to assist an authorized forensic investigator to have in-depth understanding of physiological data with multiple perspectives. In addition, our framework adopts a scalable encrypted index construction which provides provable security guarantees. Finally, we give a discussion of our future work based on this framework. As a generic and scalable framework, our design can be easily extended to secure update and parallel processing.**

Keywords: Digital Forensics, Medical Data, Time-series Data, Privacy Protection.

I. Introduction

Electronic medical record (EMR) forensics has served as an essential part of the law enforcement over the past twenty years. Analyzing EMR forensics data is at the forefront of both academia and industry, providing digital evidence to the state and federal governments, insurance companies, and private investigators [1]. However, due to the sensitive and commercially valuable data that EMR contains, data breaches in healthcare incur tremendous financial loss and put the privacy of patient in dangerous place. Therefore, mining the data by authorized entities as well as preserving data confidentiality are deemed requisite regarding the trustworthy forensic analysis.

A classical privacy-preserving technique is adopting client-side encryption to ensure end-to-end data security. But this technique restricts the functionality in forensic analyzing. A common example is detecting similar physiological trajectories for a given investigation query record over encrypted physiological data. In such case, the aforementioned solution cannot directly be applied. Rather, researchers strongly advocate approaches whereby designing cryptographic primitives with desired functionality [2], yet must nevertheless accomplish similarity match over encrypted records.

To address the above demanding needs, Kuzu *et al.* devise the first privacy-preserving similarity search algorithm over high-dimensional dataset [3]. They use the well-known locality-sensitive hashing (LSH) algorithm for finding nearest neighbors (NNs) of a given query in sublinear time [4]. By viewing LSH values as keywords, their proposed searchable encryption algorithm supports encrypted similarity search. Going a step further, Liu *et al.* design a secure and scalable similarity search service for large high-dimensional datasets [5]. Their system presents a distributed architecture which enables parallel computing and highly reduces client-side workload. Unfortunately, the above literature does not hold the ability to deal with the high-dimensional time-series medical data, such as physiological data.

The limitation is subject to the nature of LSH algorithms. Loosely speaking, the relationship between a distance function for measuring similarity and its associated LSH hash function family is a unique one-to-one mapping. To be more concrete, a single LSH hash family with its corresponding distance function can measure only one perspective on data points, such as shape, angle, and amplitude. Referred to Figure 1, we retrieve the arterial blood pressure waveforms from the MIMIC-II database [6]. When applying the LSH function based on the Cauchy distribution (ℓ_1 norm), the similarity between time-series data is mainly recognized by the amplitude of the waveforms, as (a,d) and (b,d). While (a,b) and (c,d) are considered to be similar whereby LSH for cosine distance. The reason is that the cosine distance LSH requires the waveform points to be projected onto a unit sphere, and the similarity is measured by the shape and angle.

The aim to investigate forensic physiology time-series data, however, requires interpreting waveforms via diverse perspectives. For example, of forensic consideration, the mean blood pressure, following from the measurement of amplitude, is used as a bioindicator for the presence of psychoactive drugs in the drug recognition evaluation (DRE) [1]. Likewise, the acutely and sharply increasing trend, measured by the shape of the waveform, may contain the resultant information of sudden death. In practice, measuring both facets of the similarity is highly beneficial to provide comprehensive evidence in identifying crimes.

Contributions: Having laid the intrinsic property of LSH and

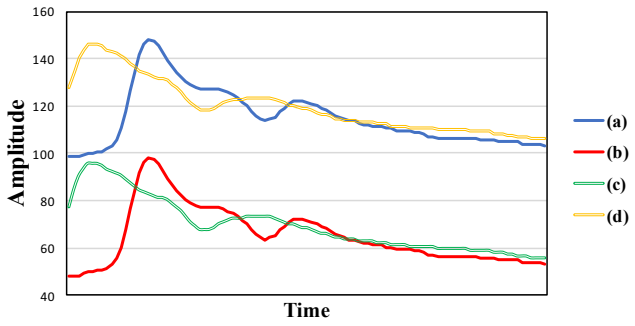

Fig. 1: An illustration of time-series ABP waveform with different amplitudes and shapes. Via the ℓ_1 distance, waveform (a,d) and (b,c) are categorized as similar, whereas via the cosine distance, (a,b) and (c,d) are recognized similar.

the requirement of medical forensics, we design a privacy-preserving forensic analysis framework for high-dimensional physiological data. Our proposed framework is capable of proceeding similarity match with diverse and refined facets over time-series data by utilizing stratified LSH (SLSH) [7]. In the meanwhile, it offers strong security guarantee for off-the-shelf databases by employing searchable symmetric encryption (SSE) techniques. As a scalable and generic framework, it can be extended to a dynamic as well as distributed system.

Our design starts from SLSH, which integrates multiple LSH hash families in terms of various distance functions. It enables a hierachical index structure for encrypted similarity search. Standing at the forensic viewpoint, we choose two level LSH families, thereby exploring physiological data with the amplitude as well as the shape and angle perspectives. By applying the outer level LSH based on ℓ_1 distance (L1LSH), we roughly group data points into big buckets by the amplitude. Then, by adopting the inner level cosine distance LSH (COSLSH) within each bucket, the data points are categorized refinedly via the shape and angle of time-series waveforms.

For privacy protection, our framework leverages SSE to provide guaranteed security strength such that the underlying contents of queries and results are only learned by authorized investigators. By merging the above two level hash values as the encrypted key, our framework prevents the storage server from learning extra information from the layered LSH index. Specifically, searched query points (aka query tokens) reveal matched encrypted results only if hash values in both layers are matched. The server will not be aware of whether the match exists in outer or inner level LSH separately. A salient feature provided by the framework is reducing the bandwidth consumption and accelerating query processing. Because the inner level LSH divides the populous outer level hash tables into small buckets, the number of result candidates fetched by the investigator will be highly lessen.

Organization: Section II investigates related literatures. Section III introduces the preliminaries used in this paper. After that, Section IV describes the workflow of our framework at high level and the proposed construction. Finally, we conclude our paper and discuss future work in Section V.

II. Related Works

Our framework is closely related to privacy-aware similarity search schemes over high-dimensional data points. The common practice is adopting LSH on top of the SSE, whereby considering the one-way transformed hashes as keywords, thus locating matching encrypted similar points. In [3], Kuzu *et al.* design the first sublinear secure similarity search scheme. Their encrypted bitmap index stems from the LSH inverted index. Later, Yuan *et al.* devise a high-performance encrypted LSH index to improve the space and query efficiency, which is the first scheme targeting on million-scale dataset [8]. Going a step further, Liu et al. design a secure, scalable and quality similarity search service for large high-dimensional datasets [5]. Their system enables parallel computing, forward-secure addition, and highly reduces client-side workload. The most recent work proposed by Yuan *et al.* investigates secure similarity join queries across two datasets [9]. Their elaboration enhances the security strength without revealing the query set distribution, and further improves the scalability via processing a small subset of query points and sharing the results with other nearby points.

The above schemes cannot handle the physiological time-series data because they only support single LSH family with a corresponding unique distance measure. Compared to the schemes above, our proposed framework stresses on interpreting similarity of physiological time-series data by diverse perspectives in a private manner. It presents a customized encrypted searchable index based on stratified LSH. Our framework is scalable and generic, and can be extended to a dynamic as well as distributed system.

III. Preliminaries

Cryptographic Primitives: A symmetric encryption scheme contains three polynomial-time algorithms (KGen,Enc,Dec). The key generation algorithm KGen is a probabilistic algorithm that takes a security parameter λ to output a secret key k; The encryption algorithm Enc is a probabilistic algorithm that takes a key k and a message $m \in \{0,1\}^*$ to output ciphertext $c \in \{0,1\}^*$; The decryption algorithm Dec is a deterministic algorithm that takes k and c to output m if c is derived from k.

Define a family of pseudo-random functions $\mathsf{F} : \{0,1\}^\lambda \times \{0,1\}^m \rightarrow \{0,1\}^n$, if for all probabilistic polynomial-time distinguishers $\mathcal{A}$, $|Pr[\mathcal{A}^{\mathsf{F}:(k,\cdot)} = 1 | k \xleftarrow{\$} \{0,1\}^\lambda] - Pr[\mathcal{A}^g = 1 | g \xleftarrow{\$} \{\mathsf{Func}[m,n]\}]| < negl(\lambda)$, where $negl(\lambda)$ is a negligible function in λ.

Locality-Sensitive Hash Family: Locality sensitive hashing [4] constructs a randomized algorithm to enable approximate yet fast similarity search in high-dimensional spaces. LSH holds the property that similar data points have hash collisions with a much higher probability than those that are far apart. A general definition of LSH is clarified below:

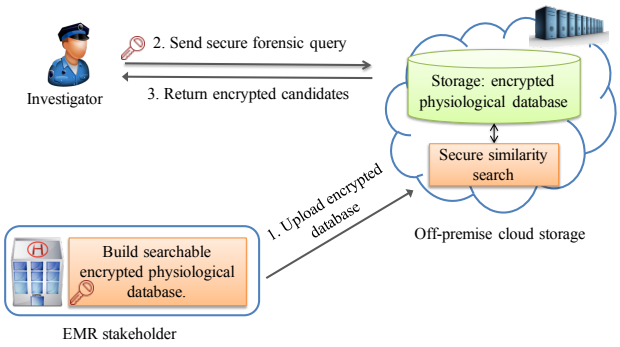

Fig. 2: The framework architecture

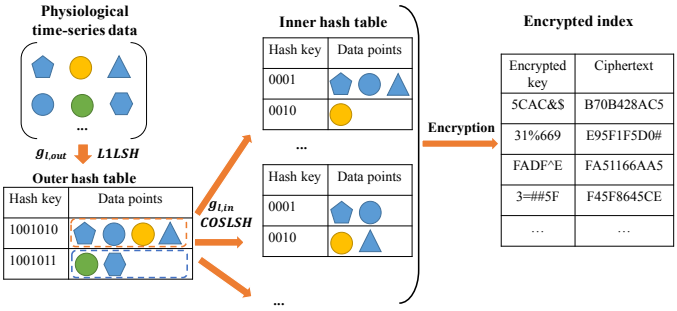

Fig. 3: The structure of the index

***Definition* 1** (Locality-sensitive Hashing)**.** *Let $\mathcal{S}$ be the domain of data points and $\mathcal{D}$ be the distance function. Given distance R_1, R_2, where $R_1 < R_2$, and probability p_1, p_2, where $p_1 > p_2$, a hashing function family $\mathcal{H} = \{h : \mathcal{S} \to \mathcal{U}\}$ is (R_1, R_2, p_1, p_2)-locality-sensitive if for any $s_i, s_j \in \mathcal{S}$: if $\mathsf{dist}(s_i, s_j) \leq R_1$ then $P[h(s_i) = h(s_j)] \geq p_1$; if $\mathsf{dist}(s_i, s_j) > R_2$ then $P[h(s_i) = h(s_j)] \leq p_2$.*

LSH algorithm concatenates multiple hash functions $h \in \mathcal{H}$ to enlarge the gap between p_1 and p_2 [4]. One composite LSH function family is defined as $\mathcal{G} = \{g : \mathcal{S} \to \mathcal{U}^m\}$, where each single composite hash function is $g(s) = (h_1(s), \cdots, h_m(s))$, and $h_i \in \mathcal{H}$. To achieve high accuracy, the algorithm independently and randomly picks L composite LSH functions from $\mathcal{G}$, and then each data point is partitioned into L buckets $\{g_1(s), \cdots, g_L(s)\}$. Given a query point q, LSH algorithm computes $\{g_1(q), \ldots, g_L(q)\}$ and collects candidates from pre-built buckets via hash matches. Finally, it evaluates the distance from each candidate to q and reports the near neighbors within a distance R or rank the candidates if required.

Consider a data point $s \in \mathbb{X}^{dim}$, the bit-sampling based LSH family for ℓ_1 distance (L1LSH) is $\mathcal{H}_{L1} = \{h : \mathbb{X}^{dim} \to \{0,1\}\}$, where $h(s) = 0$, if $s_i < t_i$ or $h(s) = 1$, if $s_i \geq t_i$. Here the i is a single dimension chosen uniformly and randomly, and s_i is the value of data point s on the i^{th} dimension, and t_i is a uniformly picked threshold upon the range of s_i. Here we leverage the parameter-free feature of bit-sampling based L1LSH, such that no tuning is required unlike other ℓ_1-norm LSH [4]. For data points $s, q \in \mathbb{X}^{dim}$, we consider the angle $\theta(s,q) = \arccos(\frac{s \cdot q}{\|s\|\|q\|})$ between them as the distance measure. As defined in [4], the LSH family for cosine distance (COSLSH) is $\mathcal{H}_{cos} = \{h : \mathbb{X}^{dim} \to \{0,1\}\}$ such that $h_r(s) = 0$, if $s \cdot r < 0$, or $h_r(s) = 1$, if $s \cdot r \geq 1$, where the projection vector $r \in \mathbb{R}^d$ is derived from every coordinate of r from isotropic Gaussian distribution $N(0,1)$.

IV. Our Proposed Framework

A. Framework Overview

This section overviews the architecture and threat model of our proposed framework. Figure 2 displays a typical forensic investigation scenario on an EMR database. A stakeholder deploys an encrypted high-dimensional physiological dataset to a server and delegates the server to support secure similarity search for an investigatory EMR query record. This server could be located at cloud as depicted or at some data center, and the EMR database is encrypted to ensure robust protection against unauthorized access.

Before proceeding, the EMR stakeholder preprocesses the physiological dataset, encompassing onset detection for finding beat duration, as well as validation. The reason to do above procedures comes from the oscillatory waveforms and noises in physiological signals, respectively. Specifically, one waveform contains multiple beat durations, and the period of beat duration varies from person to person, as well as within an individual. Therefore, it is essential to filter the noise and process data beat-by-beat [10]. To preserve data confidentiality, the stakeholder then runs the Encryption algorithm to encrypt the entire dataset and construct secure index before uploading.

After the encrypted index and data are deployed, an authorized forensic investigator is permitted to proceed secure similarity match. To start an investigation with a given query, the investigator first performs those mentioned above preprocess. Afterward, the investigator invokes the SearchToken function to derive encrypted search tokens from LSH hashes of the query. Then it sends the tokens to the server. After receiving tokens, the server runs the SimSearch function to locate, fetch, and return the result candidates. At last, the investigator decrypts and computes their distances to the query for validation and ranking purposes.

Threat Model and Security Goals: Our framework considers and withstands two kinds of adversaries. The first is the *external* adversaries. Such an adversary observes the communication and gathers static snapshots of encrypted indexes, databases, queries, and results. The second is defined as *internal* adversaries. They are able to monitor and access the memory and disks of the server. Currently, our framework does not address malicious adversaries who aim to modify the data and forensic results. Overall, our security goal is to provide reliable and guaranteed protection for sensitive EMR databases no matter they are deployed in cloud service or a private data center. Only authorized investigators can send forensic queries and obtain encrypted investigation results. The server will always see ciphertexts during the investigation procedure.

B. Construction

Design rationale: For the sake of clarity, we present the high-level idea for our framework. Our encrypted index straightfor-

Algorithm 1: Encryption$(K, \mathbf{S}, \mathbf{l1lsh}, \mathbf{coslsh})$

Input: EMR stakeholder's private key: K; Physiological dataset: $\mathbf{S} = \{s_1, \ldots, s_n\}$; SLSH parameters: $\mathbf{l1lsh}, \mathbf{coslsh}$.
Output: Encrypted Index: I; Encrypted dataset: $\mathbf{S}^*$.
begin

Initialize hash tables $H_{counter}$ and I, arrays $Array_B$ and $Array_b$;
//Phase 1: encrypted dataset;
for $i \leftarrow 1$ *to* N **do**
  $s_i^* \leftarrow \mathsf{Enc}(K, s_i)$;
  Add s_i^* to encrypted dataset $\mathbf{S}^*$;
// Phase 2: build an encrypted index;
for $\forall s \in \mathbf{S}^*$ **do**
  for $i \leftarrow 1$ *to* L_{out} **do**
    $Array_B[i] \leftarrow g_{i,out}(s) \| i$;
  for $j \leftarrow 1$ *to* L_{in} **do**
    $Array_b[j] \leftarrow g_{j,in}(s) \| j$;
  for $i \leftarrow 1$ *to* L_{out} **do**
    for $j \leftarrow 1$ *to* L_{in} **do**
      $\mathsf{K}_1 \leftarrow F_1(K, Array_B[i] \| Array_b[j])$;
      $\mathsf{K}_2 \leftarrow F_2(K, Array_B[i] \| Array_b[j])$;
      $c \leftarrow H_{counter}.\mathsf{Get}(\mathsf{K}_1)$;
      if $c \neq null$ **then**
        $H_{counter}.\mathsf{Update}(\mathsf{K}_1, c++)$;
      else
        $H_{counter}.\mathsf{Put}(\mathsf{K}_1, 1)$;
      $I.\mathsf{Put}(F_3(\mathsf{K}_1, c), \mathsf{Enc}(\mathsf{K}_2, id))$

Algorithm 2: Search()

Input: Query: s_q; Investigator's key: K; SLSH parameters: $\mathbf{l1lsh}, \mathbf{coslsh}$; Encrypted index: I; Encrypted physiological dataset: $\mathbf{S}^*$.
Output: Candidate set: R.
begin

INVESTIGATOR: SearchToken
for $i \leftarrow 1$ *to* L_{out} **do**
  $Array_B[i] \leftarrow g_{i,out}(s_q) \| i$;
for $j \leftarrow 1$ *to* L_{in} **do**
  $Array_b[j] \leftarrow g_{j,in}(s_q) \| j$;
for $i \leftarrow 1$ *to* L_{out} **do**
  for $j \leftarrow 1$ *to* L_{in} **do**
    $t_{i,j}^1 \leftarrow F_1(K, Array_B[i] \| Array_b[j])$;
    $t_{i,j}^2 \leftarrow F_2(K, Array_B[i] \| Array_b[j])$;
    $t_{i,j} \leftarrow t_{i,j}^1 \| t_{i,j}^2$;
Send $\mathbf{t} = \{t_{1,1}, \ldots, t_{L_{out}, L_{in}}\}$ to cloud server.
SERVER: SimSearch
for $\forall t_{i,j} \in \mathbf{t}$ **do**
  Parse $t_{i,j}$ to $t_{i,j}^1 \| t_{i,j}^2$;
for $i \leftarrow 1$ *to* L_{out} **do**
  for $j \leftarrow 1$ *to* L_{in} **do**
    $\mathsf{K}_1 \leftarrow t_{i,j}^1$, $\mathsf{K}_2 \leftarrow t_{i,j}^2$;
    for $c = 1$ *until* $I.\mathsf{Get}(F_3(\mathsf{K}_1, c)) = \perp$ **do**
      $id \leftarrow \mathsf{Dec}(\mathsf{K}_2, I.\mathsf{Get}(F_3(\mathsf{K}_1, c)))$;
      Put s_{id}^* to R;

wardly combines the LSH and SSE, thus enabling sublinear time search, without suffering from the "curse of dimensionality." To interpret the physiological time-series data, we measure its similarity from multiple perspectives. We adopt two LSH families bacause both the information extracted from the amplitude as well as the shape and angle from the physiological waveform is useful, of the forensic consideration.

Referred to Figure 3, from the perspective of plaintext construction, the index encompasses two strata of hash tables. At first, it groups the time-series data points via outer level L1LSH based on the amplitude. Then, by leveraging COSLSH, it finely partitions the outer level tables into smaller inner level tables. From the perspective of the encrypted index, SLSH hash values are merged as encrypted keys, and the matched key and data point (identifier) are inserted into a searchable encryption index. Therefore, the server cannot learn the underlying combination of two LSH hashes in an index. More details will be discussed later. Also, the encrypted index can be stored in a standard in-memory hash table in key-value structure, thereby performing the fast lookup.

Construction: Before describing our construction in detail, we define the notations used throughout of our paper. We define a standard hash table H which is composed of key-value pairs. Likewise, we define an array $Array$, where $Array[i]$ denotes the item at the i^{th} position of the array. Given two binary strings X and Y, $X||Y$ represents their concatenation. Our design focuses on the forensic analytic function of similarity match on high-dimensional physiological data. Given a time-series record s, s^* is the ciphertext of s. Then the data set $\mathbf{S}$ is $\{s_1, \ldots, s_n\}$, and the encrypted dataset $\mathbf{S}^*$ is defined as $\{s_1^*, \ldots, s_n^*\}$. We consider that the identifier id of a record s is also its physical address. In our algorithms, we use L1LSH as the outer level hash family, and there are overall L_{out} composite functions $\{g_{1,out}, \cdots, g_{L,out}\}$. Likewise, we apply COSLSH as the inner level hash family, and we have totally L_{in} composite functions $\{g_{1,in}, \cdots, g_{L,in}\}$.

Algorithm 1 illustrates the encryption function for the forensic physiological data records at the EMR stakeholder side. There are two phases, building the encrypted database and searchable index, respectively. Given L1LSH composite functions $g_{i,out}$ where $i \in [1, L_{out}]$, and COSLSH composite functions $g_{j,in}$ where $j \in [1, L_{in}]$, by applying on each EMR record s, hash values $g_{i,out}(s)$ and $g_{j,in}(s)$ are computed and stored in temporary arrays $Array_B$ and $Array_b$. After that, by combining each vector obtained from two

temporary arrays, token tuple $(\mathsf{K}_1, \mathsf{K}_2)$ are generated via $F_1(K, Array_B[i]||Array_b[j])$, $F_2(K, Array_B[i]||Array_b[j])$. Here we have K as the private key, F_1, F_2 as secure PRF, $i \in [1, L_{out}]$ and $j \in [1, L_{in}]$. The reason of storing above-mentioned composite hash values in two temporary arrays $Array_B$ and $Array_b$ is saving computation comsumption. Thereby, when generating overall $L_{out} \times L_{in}$ tokens, we only need to compute $(L_{out} + L_{in})$ LSH values. By employing the SSE scheme proposed in [2], each token and its associated data are constructed as unique key-value pairs to be encrypted and stored in a standard hash table. We remark that the temporary hash table $H_{counter}$ is used to cache a self-incremental counters c, such that tracking the number of matched records for each token. As a result, encrypted key-value pairs are built as $(F_3(\mathsf{K}_1, c), \mathsf{Enc}(\mathsf{K}_2, id))$, where F_3 is secure PRF.

The privacy-preserving forensic (aka Search) function is shown in Algorithm 2. An authorized investigator is obtained the key K from the EMR stakeholder to generate $L_{out} \times L_{in}$ tokens $\mathbf{t} = \{t_{1,1}, \cdots, t_{L_{out},L_{in}}\}$ from a query record s_q. Then the investigator sends them to the remote server. On the server side, the token list $\mathbf{t}$ is parsed to the token tuple $(\mathsf{K}_1, \mathsf{K}_2)$. Then the server uses token tuple to locate candidate records via $\mathsf{Dec}(\mathsf{K}_2, I.\mathsf{Get}(F_3(\mathsf{K}_1, c)))$. Due to the encrypted index and database are co-located, the server directly returns the candidates in ciphertext. After that, the investigator decrypts the candidates and computes the distances between the candidates and query record to eliminate false positives and obtain nearest neighbors if required.

Security Guarantee: The above framework provides robust security guarantee under the security of searchable symmetric encryption. Within the procedure of forensic query, the server does not see any cleartext of the datasets, queries, and results. The Encryption function reveals the total number of encrypted records n, the length of ciphertext $|s^*|$, as well as the length of key-value pairs in the index. After performing Search function, the access pattern and search pattern are captured by server as defined in [11]. The search pattern indicates the repeat between previously searched queries and the current given query. Meanwhile, the access pattern represents the resultant encrypted records of each query. Note that in secure similarity search [3], [8] built on LSH, the token set of each query contains a set of tokens deterministically generated from LSH hash values. Therefore, intersections among different token sets also indicate the similarity between query points. As our construction hides the structure of the inner and outer LSH layers in the encrypted index, the encrypted candidates will be matched only if tokens in both layers are matched. More detailed security analysis will be conducted in future work.

Remark on Correctness and Performance: The correctness is ensured regarding the deterministic mapping between the input and output of PRF. The record along with its unique LSH hash holds the corresponding matched token just like existing schemes [3], [8]. Regarding performance, the token computation overhead is subject to the number of LSH functions. As the outer level L1LSH and inner level COSLSH constitute L_{out} and L_{in} composite hash functions, such complexity is $O(L_{out} + L_{in})$. Given N records, the total cost for building the index is $O(NL_{out}L_{in})$, as each record needs to be processed by two-layer LSH functions in a nested manner. Our framework achieves asymptotic query efficiency, and the complexity is $O(|R|L_{out}L_{in})$, where $|R|$ is the size of the maximum candidate set.

V. Conclusion and Future Work

In this paper, we design a privacy-preserving framework for enabling forensic analytics on encrypted physiological data. Our construction is a generic and scalable framework for proceeding secure similarity search over high-dimensional time-series medical data. We adopt the SLSH on top of SSE to enable sublinear time search, without suffering from the "curse of dimensionality", while interpreting the time-series waveforms by diverse perspectives. In this light, our framework embraces any LSH functions with matched distance measurements, thus dealing with other kinds of seasonal time-series data. From the performance point of view, we are going to implement our framework as distributed architecture because of the co-located encrypted index and database. Furthermore, we plan to extend our framework to support secure update.

Acknowledgment

This work was supported by the OCSC POC scheme.

References

[1] J. Brick, *Forensic Alcohol Test Evidence (FATE): A Handbook for Law Enforcement and Accident Investigation*. Charles C Thomas Publisher, 2016.

[2] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in *Proc. of NDSS*, 2014.

[3] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in *Proc. of IEEE ICDE*, 2012.

[4] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM*, vol. 51, pp. 117–122, 2008.

[5] X. Liu, X. Yuan, and C. Wang, "Encsim: An encrypted similarity search service for distributed high-dimensional datasets," in *Proc. of IEEE/ACM IWQoS*, pp. 1–10, 2017.

[6] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark, "Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database," *Critical care medicine*, vol. 39, no. 5, p. 952, 2011.

[7] Y. B. Kim, E. Hemberg, and U.-M. O'Reilly, "Stratified locality-sensitive hashing for accelerated physiological time series retrieval," in *Proc. of IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016.

[8] X. Yuan, H. Cui, X. Wang, and C. Wang, "Enabling privacy-assured similarity retrieval over millions of encrypted records," in *Proc. of ESORICS*, 2015.

[9] X. Yuan, X. Wang, C. Wang, C. Yu, and S. Nutanong, "Privacy-preserving similarity joins over encrypted data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2763–2775, 2017.

[10] A. Waldin, K. Veeramachaneni, and U.-M. O'Reilly, "Learning blood pressure behavior from large physiological waveform repositories," in *Proc. of ACM ICML Workshop*, 2013.

[11] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.