

Privacy-Preserving Identification via Layered Sparse Code Design: Distributed Servers and Multiple Access Authorization

Behrooz Razeghi, Slava Voloshynovskiy, Sohrab Ferdowsi and Dimche Kostadinov

Stochastic Information Processing Group

Department of Computer Science, University of Geneva, Switzerland

{behrooz.razeghi, svolos, sohrab.ferdowsi, dimche.kostadinov}@unige.ch

Abstract—We propose a new computationally efficient privacy-preserving identification framework based on layered sparse coding. The key idea of the proposed framework is a sparsifying transform learning with ambiguization, which consists of a trained linear map, a component-wise nonlinearity and a privacy amplification. We introduce a practical identification framework, which consists of two phases: public and private identification. The public untrusted server provides the fast search service based on the sparse privacy protected codebook stored at its side. The private trusted server or the local client application performs the refined accurate similarity search using the results of the public search and the layered sparse codebooks stored at its side. The private search is performed in the decoded domain and also the accuracy of private search is chosen based on the authorization level of the client. The efficiency of the proposed method is in *computational complexity* of encoding, decoding, “encryption” (ambiguization) and “decryption” (purification) as well as *storage complexity* of the codebooks.

Index Terms—data privacy; sparse codebook; transform learning; successive refinement; ambiguization.

I. INTRODUCTION

Privacy-preserving identification is of great importance for the growing amount of applications that require fast and accurate identification. Third parties are assumed to perform the expected services but are curious about the nature of the data content of the queries. These applications include but are not limited to the internet-of-things (IoT), biometrics, clinical reports, etc.

In this work, we propose a new distributed framework of privacy-preserving identification based on successive refinement. The successive refinement of information was first studied for the classic source coding problem [1]. The performance of this problem is formulated by a rate-distortion theory. The objective is to achieve the rate-distortion bound at each stage. In [2] the authors proposed the Sparse Ternary Coding (STC) scheme for fast search in large scale identification problems. The theoretical results of the STC scheme are studied in [3]. Inspired by the successive refinement of information problem, the authors proposed a multi-layer network which successively generates sparse ternary codes, which closely achieve the Shannon lower bound of the distortion-rate function.

This research has been supported by the ERA-Net project ID_IoT No 20CH21_167534.

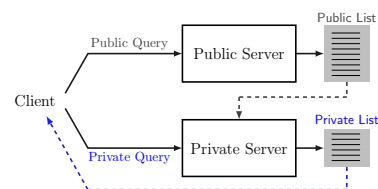


Fig. 1: General block diagram of the proposed framework.

A. Our Contribution

In this paper, we propose a new framework of multi-stage identification using successive refinement with sparse ternary codes at each layer of the privacy-preserving identification. The proposed privacy-preserving mechanism is based on the ambiguization, i.e., addition of noise to non-zero sparse data representation in the transform domain. We demonstrate that the security of this scheme does not rely on the secrecy of transform. Accordingly, we develop a distributed search framework (Fig. 1) with a granted granular access to the results of the search based on the level of authorization expressed in the knowledge of codebook and vote refinement. We demonstrate that the identification based on compressed STC representation could be a good first stage for the fast public identification, while the authorized private users can benefit from the refined results enjoying the accurate upgrades in the reconstructed real space with a low computational complexity. Up to our best knowledge, the proposed scheme is among the first that is based on the successive refinement with the sparse ternary coding bridging the gap to the theoretical performance limits.

B. Notation

Matrices and vectors are denoted by boldface upper-case (\mathbf{X}) and lower-case (\mathbf{x}) letters, respectively. We consider the same notation for a random vector \mathbf{x} and its realization. The difference should be clear from the context. x_i denotes the i -th entry of vector \mathbf{x} . For a matrix \mathbf{X} , $\mathbf{x}(j)$ denotes the j -th column of \mathbf{X} . The superscript $(\cdot)^\dagger$ stands for the pseudo-inverse and $(\cdot)^T$ stands for the transpose. We use the notation $[N]$ for the set $\{1, 2, \dots, N\}$.

C. Outline of the Paper

The remainder of the paper is organized as follows. In Sec. II, the problem formulation is introduced. Then, in Sec. III we present our framework. We provide the performance analysis in Sec. IV. Finally, conclusions are drawn in Sec. V.

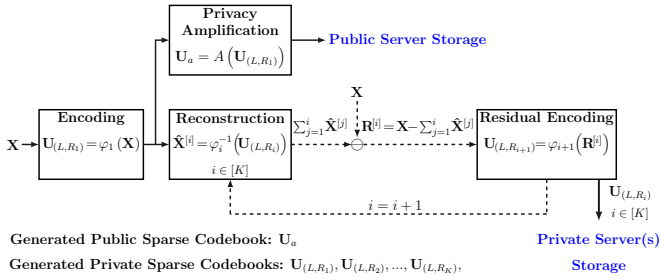


Fig. 2: Successive sparse codebooks generation scheme.

II. PROBLEM FORMULATION

Suppose that an owner has a collection of M raw vectors $\mathbf{x}(m), m \in [M]$ in the database $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(M)]$, where each raw vector $\mathbf{x}(m), m \in [M]$ from a set $\mathcal{X} \subset \mathbb{R}^N$ is a random vector with distribution $p(\mathbf{x})$ and bounded variance $\sigma_{\mathbf{x}}^2$. In general, the input data might be raw or based on extracted features such as those from (aggregated) local descriptors [4]–[6], or the top layer of a neural network [7] or the latent space of auto-encoders [8]. The user has a query $\mathbf{y}(m) \in \mathbb{R}^N$ which is a noisy version of $\mathbf{x}(m)$, i.e., $\mathbf{y}(m) = \mathbf{x}(m) + \mathbf{z}$, where we assume $\mathbf{z} \in \mathbb{R}^N$ is a Gaussian noise vector with distribution $\mathcal{N}(\mathbf{0}, \sigma_{\mathbf{z}}^2 \mathbf{I}_N)$. The user is interested in some information about the subset $\mathcal{L}(\mathbf{y})$ of the γ -NN (or γ -ANN) of \mathbf{y} . The owner subcontracts the similarity search to an entity called the server. The clients and data owner attempt at protecting their data from (public) server side analysis, which is assumed to be honest but curious.

III. PROPOSED FRAMEWORK

A. Framework Overview

Our framework is composed of the following five steps:

1) *Preparation at Owner Side*: The owner generates one *public sparse codebook* plus K *private sparse codebooks* from the media data that he owns (Fig. 2). The public codebook is sent to the public storage server (e.g., Google Site) and the K private sparse codebooks are sent to the private server storage (e.g., “Friend” Sites). The public sparse codebook is generated using the *learned sparsifying transform* followed by an *element-wise nonlinearity* and a *privacy amplification*. The K private codebooks are generated by the successive refinement encoder that will be explained in the text below.

2) *Indexing at Server Sides*: The public and private servers index the received sparse codes.

3) *Querying at Client Side*: The client generates a sparse code from his query data using the same transformation scheme used for *public* search (Fig. 3). Then, the client sends the sparse code of his query data to the public server and his original domain query to the private server.

4) *Initial Search at Public Server Side*: The server runs a similarity search to identify the sparse codes that are most similar to the query (Fig. 3). The public list, which consists of indices of the most similar codes, is sent to the private server.

5) *Multi-layer List Refinement at Private Server*: The private server looks at his first layer codebook and decodes (reconstructs) the sparse codes that are within the public list. Then he runs a similarity search using the received query and

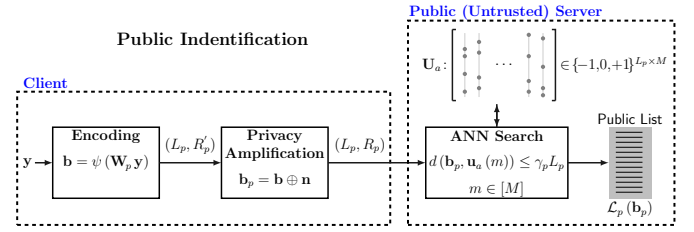


Fig. 3: Public identification scheme.

the decoded sparse codes, i.e., the similarity is computed in the original domain. This similarity search results in the first private list, which is accessible to the authorized users at level 1. Next, the private server uses his second layer codebook and decodes the sparse codes with indices within the initial private list. The second private list is hereby computed using similarity search between the received query and *superposition* of the decoded sparse codes of this layer and the previous layer. This list is accessible to the authorized users at level 2. Analogously, the private lists are refined successively by running the similarity search between the query and the superposition of the decoded sparse codes of each layer and all previous layers (Fig. 4).

B. Layered Sparse Coding

1) *Principal Element*: The core of our coding paradigm is as follows:

Encoder: This is defined by a mapping $\varphi : \mathbb{R}^N \rightarrow \{-1, 0, +1\}^L$. Given the (raw) feature vector $\mathbf{x}(m), m \in [M]$ the encoder generates the sparse code $\mathbf{u}_{(L,R)}(m), m \in [M]$ with dimensionality L and rate R , therefore $\mathbf{u}_{(L,R)}(m) = \varphi(\mathbf{x}(m))$.

Indeed, our encoder is based on the sparsifying transform learning model [9] followed by a non-linearity thresholding function to constraint the alphabet of codes. This model suggests that a feature vector $\mathbf{x}(m) \in \mathbb{R}^N$ is approximately sparsifiable using a transform $\mathbf{W} \in \mathbb{R}^{L \times N}$, that is $\mathbf{W}\mathbf{x}(m) = \mathbf{a}(m) + \mathbf{e}_a$, where $\mathbf{a}(m) \in \mathbb{R}^L$ is sparse, i.e., $\|\mathbf{a}(m)\|_0 \ll L$, and $\mathbf{e}_a \in \mathbb{R}^L$ is the representation error of the feature vector or residual in the *transform domain*. The sparse coding problem for this model is a direct constraint projection problem. This sparse approximation is as follows:

$$\hat{\mathbf{a}}(m) = \arg \min_{\mathbf{a}(m) \in \mathbb{R}^L} \|\mathbf{W}\mathbf{x}(m) - \mathbf{a}(m)\|_2^2 + \lambda \Omega(\mathbf{a}(m)), \forall m \in [M].$$

The above direct problem has a closed-form solution for any of the two important regularizers $\Omega(\cdot) = \|\cdot\|_0$ or $\Omega(\cdot) = \|\cdot\|_1$. Analogous to [10], we consider the ℓ_0 -“norm” as our sparsity-inducing penalty. In this case, the solution $\hat{\mathbf{a}}(m)$ is obtained exactly by hard-thresholding the projection $\mathbf{W}\mathbf{x}(m)$ and keeping the S_x entries of the largest magnitudes while setting the remaining low magnitude entries to zero. For this purpose, we define an intermediate vector $\mathbf{f}(m) \triangleq \mathbf{W}\mathbf{x}(m) \in \mathbb{R}^L$ and denote by λ_x the S_x -th largest magnitude amongst the set $\{|f_1(m)|, \dots, |f_L(m)|\}$. Then the closed-form solution is achieved by applying a hard-thresholding operator to $\mathbf{f}(m)$, which is defined as $\mathbf{a}(m) = H_{\lambda_x}(\mathbf{f}(m)) = \mathbb{1}_{|f_l(m)| \geq \lambda_x} \mathbf{f}(m), \forall m \in [M], \forall l \in [L]$. Now

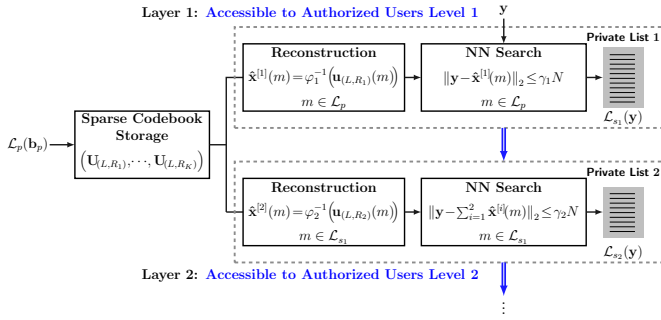


Fig. 4: Private multiple access identification scheme.

we impose extra constraint on the alphabet of our codes by applying the ternary hash mapping to $H_{\lambda_x}(\mathbf{W}\mathbf{x}(m))$ as:

$$\mathbf{u}(m) \triangleq \psi_{\lambda_x}(\mathbf{W}\mathbf{x}(m)) \in \{-1, 0, +1\}^L, \forall m \in [M], \quad (1)$$

where $\psi_{\lambda_x}(\mathbf{W}\mathbf{x}(m)) = \text{sign}(H_{\lambda_x}(\mathbf{W}\mathbf{x}(m)))$. The bit rate of this code can be formulated as $R = \frac{1}{L} \log_2 \binom{L}{S_x} 2^{S_x}$. We denote by $\varphi(\cdot)$ the encoder in general, therefore, the codeword $\mathbf{u}(m)$ with block-length L and rate R is denoted by $\mathbf{u}_{(L,R)}(m) = \varphi(\mathbf{x}(m))$, $m \in [M]$.

In general, we have a joint learning problem that can be formulated as:

$$(\hat{\mathbf{W}}, \hat{\mathbf{A}}) = \arg \min_{(\mathbf{W}, \mathbf{A})} \|\mathbf{W}\mathbf{X} - \mathbf{A}\|_F^2 + \beta_W \Omega_W(\mathbf{W}) + \beta_A \Omega_A(\mathbf{A}), \quad (2)$$

where $\beta_W \geq 0$ and $\beta_A \geq 0$ are regularization parameters, $\Omega_W(\mathbf{W})$ and $\Omega_A(\mathbf{A})$ are the constraints on the linear mapper \mathbf{W} and sparse (but not ternarized) code matrix \mathbf{A} , respectively. The algorithm for the above problem alternates between solving for $\mathbf{A} = H_{\lambda}(\mathbf{W}\mathbf{X})$ (sparse coding step) and $\mathbf{W} = \mathbf{U}_W \mathbf{V}_W^T$ (transform update step), whilst the other variables are kept fixed. Finally, the ternarized sparse codebook $\mathbf{U}_{(L,R)}$ is obtained as $\mathbf{U}_{(L,R)} = \varphi(\mathbf{X})$, which consists of M sparse codewords $\mathbf{u}_{(L,R)}(m) \in [M]$.

Decoder: This is a mapping $\varphi^{-1}: \{-1, 0, +1\}^L \rightarrow \mathbb{R}^N$. Base on $\mathbf{u}_{(L,R)}(m)$ generated at the encoder, the decoder produces reconstruction $\hat{\mathbf{x}}(m) = \varphi^{-1}(\mathbf{u}_{(L,R)}(m)) = \mathbf{W}^\dagger \mathbf{u}_{(L,R)}(m)$. That is, our decoding is simply a pseudo-inverse operation.

2) Overall Scheme: In [11], the authors studied the reconstruction performance of STC and scaled STC based on distortion-rate function. It is shown that for relatively small rates the ternarized sparsified codes almost achieve the Shannon distortion-rate function for i.i.d. Gaussian distributed data. In [12], the authors extended the concept of STC to multi-layer STC, a codebook-free scheme that successively refines the reconstruction of the residuals of previous layers. Based on the results in [11] and [12] we formulate our layered sparse coding that provides a multiple-access privacy-preserving identification scheme. To this end, first we generate our first codebook with block-length L and rate R_1 as: $\mathbf{U}_{(L,R_1)} = \varphi_1(\mathbf{X}) = \psi_{\lambda_1}(\mathbf{W}_1 \mathbf{X})$. Next, we do reconstruction as: $\hat{\mathbf{X}}^{[1]} = \varphi_1^{-1}(\mathbf{U}_{(L,R_1)}) = \mathbf{W}_1^\dagger \mathbf{U}_{(L,R_1)}$. This provides the residual $\mathbf{R}^{[1]} = \mathbf{X} - \hat{\mathbf{X}}^{[1]}$. Now, we encode the

residual of the first layer to generate the second codebook with block-length L and rate R_2 as: $\mathbf{U}_{(L,R_2)} = \varphi_2(\mathbf{R}^{[1]}) = \psi_{\lambda_2}(\mathbf{W}_2 \mathbf{R}^{[1]})$. The reconstructed data as well as the residual of the second layer obtained as $\hat{\mathbf{X}}^{[2]} = \varphi_2^{-1}(\mathbf{U}_{(L,R_2)})$ and $\mathbf{R}^{[2]} = \mathbf{X} - (\hat{\mathbf{X}}^{[1]} + \hat{\mathbf{X}}^{[2]})$, respectively. In the same way, the layered sparse coding scheme, initialized with $i = 0$, can be formulated as:

$$\begin{aligned} \mathbf{U}_{(L,R_{i+1})} &= \varphi_{i+1}(\mathbf{R}^{[i]}), \\ \hat{\mathbf{X}}^{[i]} &= \varphi_i^{-1}(\mathbf{U}_{(L,R_i)}), \\ \mathbf{R}^{[i]} &= \mathbf{X} - \sum_{j=1}^i \hat{\mathbf{X}}^{[j]}. \end{aligned} \quad (3)$$

Note that $\hat{\mathbf{X}}^{[1]} \rightarrow \dots \rightarrow \hat{\mathbf{X}}^{[K]}$ forms a Markov chain. The algorithm successively refines the original database \mathbf{X} over (asymptotically large) K stages, such that $\|\mathbf{X} - \sum_{i=1}^K \hat{\mathbf{X}}^{[i]}\|_F^2 \leq ND$.

C. Privacy Amplification Scheme

The core idea of our privacy amplification scheme is to increase the general entropy of our sparse codes via adding some randomness to it. To this end, let $\mathcal{U}, \mathcal{V} \subseteq \mathcal{T}$ be two subspaces such that $\mathcal{T} = \mathcal{U} + \mathcal{V}$, where \mathcal{T} is the space of L -dimensional sparse codes. So, every vector $\mathbf{u}_a \in \mathcal{T}$ has at least one expression as $\mathbf{u}_a = \mathbf{u} + \mathbf{v}$, $\mathbf{u} \in \mathcal{U}$, $\mathbf{v} \in \mathcal{V}$. If we have $\mathcal{U} \cap \mathcal{V} = \{\mathbf{0}\}$, then every vector $\mathbf{u}_a \in \mathcal{T}$ has the *unique* expression $\mathbf{u}_a = \mathbf{u} + \mathbf{v}$, $\mathbf{u} \in \mathcal{U}$, $\mathbf{v} \in \mathcal{V}$ and we write $\mathcal{T} = \mathcal{U} \oplus \mathcal{V}$. Also, \mathcal{T} is called the direct sum of \mathcal{U} and \mathcal{V} . Now, let \mathcal{U} be the space of non-zero components of \mathcal{T} and \mathcal{V} be the space of zero components of \mathcal{T} . The idea of our ambiguity scheme is to set ambiguity noise \mathbf{n} such that $\mathbf{n} \in \mathcal{V}$. Furthermore, since $(\mathcal{T}, \langle \cdot, \cdot \rangle)$ is an inner product space and $\mathcal{V} = \mathcal{U}^\perp \triangleq \{\mathbf{v} \in \mathcal{V} : \langle \mathbf{u}, \mathbf{v} \rangle = 0, \forall \mathbf{u} \in \mathcal{U}\}$, $\mathcal{T} = \mathcal{U} \oplus \mathcal{U}^\perp$ is orthogonal direct sum of \mathcal{U} and \mathcal{U}^\perp . It is clear that $\dim \mathcal{T} = L$, $\dim \mathcal{U} = S_x$ and $\dim \mathcal{U}^\perp = L - S_x$. For more details about the performance of ambiguity scheme we refer the reader to [10].

1) Owner's Privacy Amplification: Based on our definition, the data owner simply adds random samples with alphabet $\{-1, +1\}$ to the zero-components of his sparse codebook $\mathbf{U}_{(L,R)}$ and sends the ambiguized sparse codebook $\mathbf{U}_{(L,R_p)}$ to the public server. We denote by S_{n_s} the sparsity level of ambiguity noise at the public server. Note that $0 \leq S_{n_s} \leq L - S_x$. Furthermore, the owner may send only a fraction $L_p < L$ of his sparse codes to the public server. In [11], the authors analyzed this scheme with more details. In general, the public ambiguized sparse codebook generated as $\mathbf{U}_{(L_p,R_p)} = A(\mathbf{U}_{(L,R)})$ with the block-length L_p and the rate R_p , where A is an ambiguity function, which consists of randomness addition as well as codeword subspace selection.

2) Client's Privacy Amplification: In order to prevent reconstruction of exact information about the client's interests at the public server side, the client ambiguizes his sparse code \mathbf{b} by adding S_{n_a} random samples with alphabet $\{-1, +1\}$ to the zero-components of his query. We denote by $\mathbf{b}_p = \mathbf{b} \oplus \mathbf{n}$ the public query.

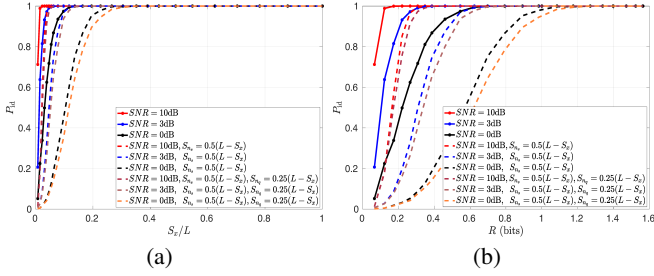


Fig. 5: The relation between probability of correct identification and a) sparsity ratio, b) encoding rate.

D. Algorithm

1) *Preparation at Owner Side*: The owner generates offline the sparse codebook $\mathbf{U}_{(L,R_1)}$ with the *trained linear map* \mathbf{W}_1 followed by the *element-wise nonlinearity* thresholding operator $\psi_{\lambda_1}(\cdot)$, i.e., $\mathbf{U}_{(L,R_1)} = \psi_{\lambda_1}(\mathbf{W}_1\mathbf{X}) = \varphi_1(\mathbf{X})$. Then, the owner performs the privacy amplification on codebook $\mathbf{U}_{(L,R_1)}$ to generate the public sparse codebook $\mathbf{U}_a = A(\mathbf{U}_{(L,R_1)})$ with block-length L_p and rate R_p . This ambiguized codebook is outsourced to the public server storage. Next, the owner generates successively the sparse codebooks $\mathbf{U}_{(L,R_i)}$, $i = 2, \dots, K$ from the first sparse codebook $\mathbf{U}_{(L,R_1)}$. Therefore, the database \mathbf{X} is encoded by total rate $R_1 + R_2 + \dots + R_K$. The K sparse codebooks $\mathbf{U}_{(L,R_i)}$, $i \in [K]$ are outsourced to the private server storage. The block diagram of codebooks generation is illustrated in Fig. 2.

2) *Indexing at Server Sides*: The public and private servers index the received sparse codes. It can be indexed as in [2].

3) *Querying at Client Side*: The client generates the sparse codeword \mathbf{b} from its query \mathbf{y} , using the shared *public trained linear map* \mathbf{W}_1 followed by the *element-wise nonlinearity* operator $\psi_{\lambda_1}(\cdot)$, therefore $\mathbf{b} = \psi_{\lambda_1}(\mathbf{W}_1\mathbf{y})$. Then, the client ambiguizes his code by adding S_{n_q} random samples with alphabet $\{-1, +1\}$ to his code. The ambiguized public query $\mathbf{b}_p = \mathbf{b} \oplus \mathbf{n}$ is send to the public server. The client also sends his original domain query \mathbf{y} to the private server. Each client has a pre-defined authorization level at the private sever.

4) *Initial Search at Public Server Side*: The public server seeks all $\{\mathbf{u}_a(m), m \in [M]\}$ ANNs in the radius $\gamma_p L_p$ from the query \mathbf{b}_p in order to produce an initial public list \mathcal{L}_p of possible candidates as $\mathcal{L}_p(\mathbf{b}_p) = \{m \in [M] : d_{A_p}(\mathbf{u}_a(m), \mathbf{b}_p) \leq \gamma_p L_p\}$, where $d_{A_p}(\cdot, \cdot)$ is a similarity measure in space \mathcal{A}_p . Next, the public server sends the initial list \mathcal{L}_p to the private server.

One can use different similarity measures $d_{A_p}(\cdot, \cdot)$. However, due to many interesting properties, we consider a new similarity and dissimilarity measures based on the support intersection of the sparse codewords [3], [13]. To this end, we decompose sparse codes into positive part and negative part as $\mathbf{u}(m) = \mathbf{u}^+(m) + \mathbf{u}^-(m)$ and $\mathbf{b}_p = \mathbf{b}_p^+ + \mathbf{b}_p^-$, where $\mathbf{u}^+(m) = \max(\mathbf{u}(m), \mathbf{0})$ and $\mathbf{b}_p^+ = \max(\mathbf{b}_p, \mathbf{0})$ correspond to positive components and $\mathbf{u}^-(m) = \max(-\mathbf{u}(m), \mathbf{0})$ and $\mathbf{b}_p^- = \max(-\mathbf{b}_p, \mathbf{0})$ correspond to negative components. The similarity score $\text{Sim}^{(m)}$ between $\mathbf{u}(m)$ and \mathbf{b}_p is defined as:

$$\text{Sim}^{(m)} = \|\mathbf{u}^+(m) \odot \mathbf{b}_p^+\|_1 + \|\mathbf{u}^-(m) \odot \mathbf{b}_p^-\|_1, \quad (4)$$

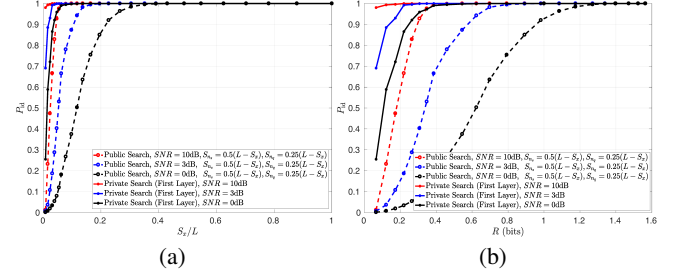


Fig. 6: Comparison between the probability of correct identification at the public server and private server.

and the dissimilarity score $\text{Dis}^{(m)}$ between $\mathbf{u}(m)$ and \mathbf{b}_p is defined as:

$$\text{Dis}^{(m)} = \|\mathbf{u}^+(m) \odot \mathbf{b}_p^-\|_1 + \|\mathbf{u}^-(m) \odot \mathbf{b}_p^+\|_1, \quad (5)$$

where \odot is the Hadamard product. For more details about the theoretical aspects of the considered similarity measure, we refer the reader to [13].

The public list \mathcal{L}_p is composed of the indices whose similarity score $\text{Sim}^{(m)}$, $m \in [M]$ is higher than a threshold and dissimilarity score $\text{Dis}^{(m)}$, $m \in [M]$ is below a threshold. Another option is to define a *normalized similarity* as $\nu^{(m)} = \text{Sim}^{(m)} / (\text{Sim}^{(m)} + \text{Dis}^{(m)})$, $\forall m \in [M]$. Therefore, the public list \mathcal{L}_p is composed of the indices of the γ largest $\nu^{(m)}$'s. Finally, the public server sends back the public list \mathcal{L}_p to the private server. The public server can either fix the threshold or the number of γ similar elements.

5) *Multiple Access List Refinement at Private Server*: The private server receives the public list \mathcal{L}_p , then it considers first layer codebook $\mathbf{U}_{(L,R_1)}$, which is the clean and full length version of the public codebook \mathbf{U}_a . Next, the private server reconstructs the codewords with indices reported on the public list as $\hat{\mathbf{x}}^{[1]}(m) = \varphi_1^{-1}(\mathbf{u}_{(L,R_1)}(m))$, $m \in \mathcal{L}_p$. It then computes the distance measure between private query \mathbf{y} and reconstructed sparse codewords $\hat{\mathbf{x}}^{[1]}(m)$, $m \in \mathcal{L}_p$ in the original signal domain. This will produced the first private list $\mathcal{L}_{s_1}(\mathbf{y}) = \{m \in \mathcal{L}_p : \|\mathbf{y} - \hat{\mathbf{x}}^{[1]}(m)\|_2 \leq \gamma_{s_1} N\}$. Next, he reconstructs the codewords $\mathbf{u}_{(L,R_2)}(m)$, $m \in \mathcal{L}_{s_1}$ of the second layer codebook $\mathbf{U}_{(L,R_2)}$. The second private list is obtained as $\mathcal{L}_{s_2}(\mathbf{y}) = \{m \in \mathcal{L}_{s_1} : \|\mathbf{y} - (\hat{\mathbf{x}}^{[1]}(m) + \hat{\mathbf{x}}^{[2]}(m))\|_2 \leq \gamma_{s_2} N\}$. In the same approach, at the K -th layer the private list is given as $\mathcal{L}_{s_K}(\mathbf{y}) = \{m \in \mathcal{L}_{s_{K-1}} : \|\mathbf{y} - \sum_{i=1}^K \hat{\mathbf{x}}^{[i]}(m)\|_2 \leq \gamma_{s_K} N\}$.

IV. PERFORMANCE ANALYSIS

In this section we analyze the performance of our method in the terms of probability of correct identification P_{id} as well as privacy leakage. To this end we consider a database \mathbf{X} of $M = 100\text{K}$ random vectors with dimensionality $N = 256$, which are generated from the distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We then generate the noisy version of \mathbf{X} with three different signal-to-noise-ratios (SNRs) 10dB, 3dB and 0dB, where $\text{SNR} = 10 \log_{10} \frac{1}{\sigma_z^2}$. We consider square sparsifying transform, i.e., $L = N$. Moreover, the sparsity level of the public sparse codewords as well as the public query code are considered the same.

In Fig. 5, we depict the probability of correctly identifying the true query in the public list \mathcal{L}_p as the function of sparsity ratio S_x/L and encoding rate R . The red, blue and black solid

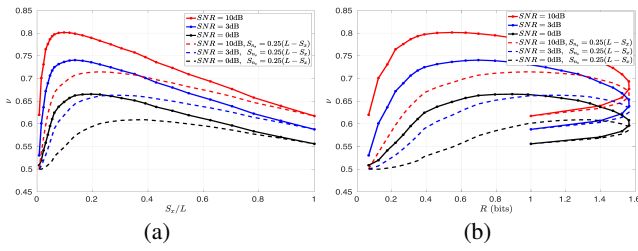


Fig. 7: The relation between normalized similarity and: a) sparsity ratio, b) encoding rate.

lines show the performance of our method in the case that we impose no privacy amplification for the stored public database and the client’s query. Next, we ambiguized our sparse public codebook by adding $S_{n_s} = 0.5(L - S_x)$ random samples in the co-support of the public codewords. Finally, we complete our scenario by considering the privacy protection of query as well as owner’s database, i.e., we ambiguized our codes by adding $S_{n_q} = 0.25(L - S_x)$ samples in the co-support of the public query codewords.

In Fig. 6, we compare the probability of correct identification at the public and private servers. We set the privacy amplification parameters of public codebook and client’s query as $S_{n_s} = 0.5(L - S_x)$ and $S_{n_q} = 0.25(L - S_x)$, respectively. Then, we perform fast public search in the transform domain and send back the public list \mathcal{L}_p to the private server. The results demonstrate high performance just by one layer similarity search in the original domain.

Based on the defined similarity measure in (4) and (5), in Fig. 7, we illustrate the relation between *normalized similarity* ν and sparsity ratio S_x/L and encoding rate R . As it is shown, at the sparsity ratios (rates) close to zero the similarity measure grows much faster than dissimilarity measure such that we have the maximal normalized similarity for relatively small sparsity ratios (rates). However, after a certain level the dissimilarity measure grows faster than the similarity measure.

In [10] and [11], we defined the privacy measures in the terms of ‘reconstruction leakage’ and ‘clustering leakage’. Based on the results in [10] and [11], the curious public server cannot perform clustering the stored public database. Moreover, the un-authorized clients cannot infer the structure of database. In order to address the reconstruction leakage of the proposed privacy-preserving identification scheme, consider the mutual information between the random sequence \mathbf{x} , the reconstructed random sequence $\hat{\mathbf{x}}_a$ at the public server and the reconstructed random sequences $\hat{\mathbf{x}}^{[i]}$, $i \in [K]$ at the private server. Using the data-processing inequality and considering the markovity of random sequences, we have:

$$I(\mathbf{x}; \hat{\mathbf{x}}_a) \leq I(\mathbf{x}; \hat{\mathbf{x}}^{[1]}) \leq \dots \leq I(\mathbf{x}; \hat{\mathbf{x}}^{[1]}, \dots, \hat{\mathbf{x}}^{[K]}).$$

In Fig. 8, we illustrate the distortion-rate behavior at the public and private servers, which interprets the ‘reconstruction’ leakage in these scenarios. Fig. 8a depicts the reconstruction leakage for three different ambiguity levels at the public sparse codebook and compares them with the Shannon lower bound. In Fig. 8b, we illustrate the performance of reconstruction at the private server and compare it with Shannon lower

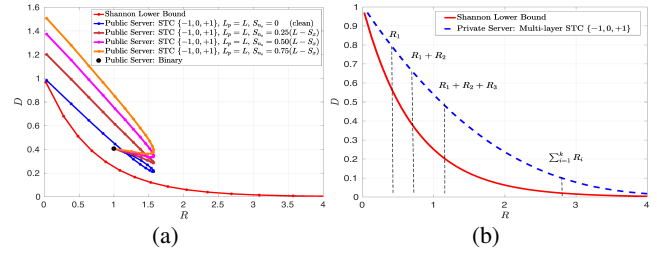


Fig. 8: Distortion-rate behavior at the a) public server, b) private server.

bound. This plot also depicts the accuracy of private lists for different authorization levels. Note that the illustrated results are obtained without considering any optimal rate allocation to our codebooks. By utilizing the optimal rate allocation and also multi-level quantization we can closely achieve the Shannon lower bound. That is beyond the scope of this paper.

V. CONCLUSION

We have proposed a novel distributed privacy-preserving identification framework based on layered sparse codes with the ambiguization and granular access to the results of identification. The initial fast search is performed on the public server and the refined searches are performed on the distributed private server(s). The accuracy of the private search is based on the authorization level of the clients. The results show the performance of proposed scheme in the terms of probability of correct identification as well as the privacy leak measures.

REFERENCES

- [1] W. H. Equitz and T. M. Cover, “Successive refinement of information,” *IEEE Transactions on Information Theory*, vol. 37, pp. 269–275, 1991.
- [2] S. Ferdowsi, S. Voloshynovskiy, D. Kostadinov, and T. Holotyak, “Fast content identification in high-dimensional feature spaces using sparse ternary codes,” in *IEEE Int. Work. on Inf. Forensics and Security (WIFS)*, 2016, pp. 1–6.
- [3] —, “Sparse ternary codes for similarity search have higher coding gain than dense binary codes,” in *IEEE Int. Symp. on Inf. Theory (ISIT)*, 2017.
- [4] H. Jégou, M. Douze, and C. Schmid, “On the burstiness of visual elements,” in *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 2009, pp. 1169–1176.
- [5] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 2007, pp. 1–8.
- [6] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 2010, pp. 3304–3311.
- [7] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *Europ. Conf. on Comp. Vision*, 2014, pp. 584–599.
- [8] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *International Conference on Learning Representations (ICLR)*, 2014.
- [9] S. Ravishanker and Y. Bresler, “Learning sparsifying transforms,” *IEEE Trans. on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, 2013.
- [10] B. Razeghi, S. Voloshynovskiy, D. Kostadinov, and O. Taran, “Privacy preserving identification using sparse approximation with ambiguization,” in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Rennes, France, December 2017, pp. 1–6.
- [11] B. Razeghi and S. Voloshynovskiy, “Privacy-preserving outsourced media search using secure sparse ternary codes,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, Calgary, Canada, April 2018, pp. 1992–1996.
- [12] S. Ferdowsi, S. Voloshynovskiy, and D. Kostadinov, “A multi-layer network based on sparse ternary codes for universal vector compression,” *ArXiv e-prints*, Oct 2017.
- [13] D. Kostadinov and S. Voloshynovskiy, “Learning non-linear transform with discriminative and minimum information loss priors,” 2018. [Online]. Available: <https://openreview.net/pdf?id=SJMzJEq6W>