Single-Component Privacy Guarantees in Helper Data Systems and Sparse Coding with Ambiguation

Behrooz Razeghi*, Taras Stanko†, Boris Škorić†, Slava Voloshynovskiy*

*Department of Computer Science, University of Geneva, Switzerland

{behrooz.razeghi, svolos}@unige.ch

†Department of Mathematics and Computer Science, Eindhoven University of Technology, the Netherlands

{t.stanko, b.skoric}@tue.nl

Abstract—We investigate the privacy of two approaches to (biometric) template protection: Helper Data Systems and Sparse Ternary Coding with Ambiguation. In particular, we focus on a privacy property that is often overlooked, namely how much leakage exists about one specific binary property of one component of the feature vector. This property is e.g. the sign or an indicator that a threshold is exceeded.

We provide evidence that both approaches are able to protect such sensitive binary variables, and discuss how system parameters need to be set.

Index Terms—privacy, biometric authentication, template protection

I. INTRODUCTION

A. Privacy-preserving storage of biometric enrollment data

Biometric data such as fingerprints and irises cannot be treated as a secret. After all, we leave latent fingerprints on many objects that we touch, and high-resolution photos of faces reveal a lot about our irises. Nonetheless, person authentication based on biometrics is still possible, provided that the verifier performs good liveness detection. In spite of the not-really-secret nature of biometric data there are very good reasons to treat them as confidential. Storing biometric databases in unprotected form would lead to various privacy issues. In this paper we focus on one particular privacy problem: some biometric data reveal medical conditions.

The protection of this kind of data must be as good as the protection of passwords. The attacker model in the case of password storage states that the adversary is an *insider*, i.e., somebody who has access to cryptographic keys. Furthermore, the standard use case considered in most of the literature dictates that the biometric prover does not have to type long keys or to present a smartcard. This combination of attacker model and use case implies that simply encrypting the confidential data is not an option. The typical solution for passwords is to apply a one-way function and to store the hash of each password. However, this solution does not work for noisy data such as biometrics; one bit flip in the input of the hash function causes 50% bit flips in the output.

Several techniques have been developed for securely storing noisy credentials, also known as template protection, in the above given context: (i) Helper Data Systems (HDS), also known as fuzzy commitment, secure sketch, fuzzy extractor [1]–[4]; (ii) Locality Sensitive Hash (LSH) functions [5], [6]; (iii) homomorphic encryption [7], [8]; and most recently (iv) Sparse Coding with Ambiguation (SCA) [9]–[12].

B. Comparison of Template Protection Techniques

The LSH approach is fast but does not give clear privacy guarantees. Homomorphic encryption has excellent privacy, but is computationally expensive. In this paper we will not consider the LSH and homomorphic crypto approach.

The HDS approach is the oldest and is well studied. Nevertheless, the narrow privacy question of *protecting one specific aspect* of the biometric, which is relevant for the above mentioned medical condition, has not been studied in detail. The aim of this paper is to compare the privacy properties of the HDS and the SCA approach, in particular the 'medical condition' aspect. Here it is important to note that previous work on SCA has focused only on the inability of an adversary to reconstruct the *full biometric* from the enrollment data; that is *not* the property we will be looking at in the current paper. Mostly, in the literature, the protection of a vector $\mathbf{x} \in \mathbb{R}^N$ is considered. However, often it is the projection of $\mathbf{x}$ onto some fixed direction $\mathbf{v}$ that is relevant, $z = \mathbf{x} \cdot \mathbf{v}$. The range in which z lies can be privacy-sensitive, e.g. the sign of z or whether z is far away from average.

C. Contributions

We concentrate on one component x_n of a to-be-protected random vector $\mathbf{x}$, in particular a binary property $\psi(x_n)$, which is either the sign or an 'extremeness' indicator that checks if $|x_n|$ exceeds some threshold. We investigate how much information leaks about $\psi(x_n)$ through the enrollment data.

- In quantizing HDSs high leakage can occur if a bad parameter choice is made. The best choice is to take an *even* number of quantization intervals, and to subdivide them into *two* helper data intervals; then there is zero leakage about the sign and the 'extremeness'.
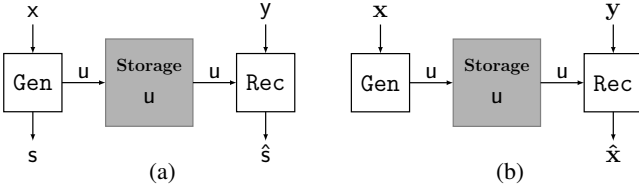
Fig. 1: *Data flow in: (a) generic Helper Data System and (b) general Sparse Coding with Ambiguation mechanism.*
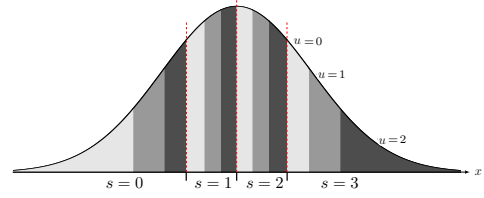

Fig. 2: *Example of the Zero Leakage discretising HDS with four quantisation intervals $s \in \{0, 1, 2, 3\}$. The discrete helper data $u \in \{0, 1, 2\}$ is indicated as grayscales.*

- The Code Offset Method causes negligible leakage.
- In the SCA mechanism, leakage about the 'extremeness' bit can be made negligibly small by setting the ambiguation noise level larger than the Hamming weight of the sparse ternary representation. This noise has little impact on the performance of the authentication system, since it gets removed in case of a genuine user's verification measurement.

II. PRELIMINARIES

A. Notation and Terminology

Vectors and matrices are denoted by boldface lower-case ($\mathbf{v}$) and upper-case ($\mathbf{M}$). When there is no distinction between a scalar (w), vector ($\mathbf{w}$), or matrix ($\mathbf{W}$), we write w. The notation $\mathbf{m}(j)$ denotes the j-th column of $\mathbf{M}$. An enrollment measurement will be written as a vector $\mathbf{x}$, and the verification measurement as $\mathbf{y}$. When a distinction needs to be made between a random variable (RV) and its numerical value, the RV is written in capitals, and the value in lowercase. Expectation over x is denoted as $\mathbb{E}_x$. The Shannon entropy of a discrete RV X is denoted as $H(X)$ and is defined as $H(X) = \sum_x p_x \log \frac{1}{p_x}$. The conditional entropy of X given Y is written as $H(X|Y)$ and is defined as $H(X|Y) = \mathbb{E}_y H(X|Y=y)$. The mutual information between X and Y is $I(X;Y) = H(X) - H(X|Y)$. For $p \in [0, 1]$ the binary entropy function h is defined as $h(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}$. We use the notation $[n] = \{1, ..., n\}$. The superscript $(\cdot)^T$ stands for the transpose and $(\cdot)^\dagger$ for pseudo-inverse. The logarithm 'log' has base 2. Bitwise XOR is denoted as $\oplus$. The Heaviside step function is written as $\Theta(\cdot)$.

We will work with the following *authentication* setting. The Verifier owns an enrolment database of public data $P(c)$ for a set of users $c \in [C]$. In the verification phase he is presented with a vector $\mathbf{y}$ and a user label c; his task is to decide if $\mathbf{y}$ is consistent with $P(c)$.

B. Zero Leakage Helper Data Systems

A HDS in its most general form is shown in Fig. 1a. The `Gen` procedure takes as input a measurement x. `Gen` outputs a secret s and public Helper Data u, where x, s and u can be scalar, vector, or matrix, in general. The helper data is stored in a memory that can be accessed by the adversary. In the reproduction phase, a fresh measurement y is obtained. Typically y is close to x but not identical. The `Rec` procedure takes y and u as input. It outputs ŝ, an estimate of s. If y is sufficiently close to x, then ŝ = s. The 'Zero Leakage' (ZL) property is defined as $I(U;S) = 0$, i.e., the helper data

reveals nothing about the secret. Obviously, U has to leak about X, since U is a function of X. When s = x the HDS is referred to as a Secure Sketch. When S given U has a uniform distribution, the HDS is called a Fuzzy Extractor. If X is a continuum variable, the first step in the signal processing is discretisation. For this purpose, a special ZLHDS has been designed [13]–[15] which reduces quantisation errors. The distribution of X needs to be known. After discretisation the Code Offset Method can be applied (Section II-C).

The discretising ZLHDS is shown in Fig. 2. Consider a source $X \in \mathbb{R}$, with $X \sim f(x)$. Let F be the cumulative distribution. The x-axis is divided into J quantisation intervals corresponding to the extracted secret $s \in \{0, 1, \ldots, J-1\}$. The distribution of S is not necessarily uniform. Let q_s be the left boundary of the interval $S = s$. Let $\{p_s\}_{s=0}^{J-1}$ denote the probabilities of the s-values. Then $F(q_s) = \sum_{i=0}^{s-1} p_i$. Each s-interval is equiprobably divided into m sub-intervals (depicted as grayscales in Fig. 2); the helper data u is defined as the index of the sub-interval in which the enrollment x lies. The s and u are computed from x as follows:

$$s = \max\{t | q_t \le x\} \quad ; \quad u = \left\lfloor m \frac{F(x) - F(q_s)}{p_s} \right\rfloor . \quad (1)$$

In the limit $m \to \infty$, the helper data can be seen as a *quantile* $\tilde{u} = \frac{u}{m} \in [0, 1)$ within the s-interval. It holds that $F(x) = F(q_s) + \tilde{u} p_s$.

C. The Code Offset Method [1], [16]

Consider $\psi(\mathbf{x}), \psi(\mathbf{y}) \in \{0, 1\}^N$. Consider a linear binary error-correcting code with syndrome function `Syn` : $\{0, 1\}^N \to \{0, 1\}^{N-k}$ and syndrome decoding function `SynDec` : $\{0, 1\}^{N-k} \to \{0, 1\}^N$, where k is the message length. The Code Offset Method in its simplest form can be used as a Secure Sketch for any distribution of X. The helper data and the reconstruction $\widehat{\psi(\mathbf{x})}$ are defined as follows:

$$\mathbf{u} = \text{Syn}\,\psi(\mathbf{x}); \quad \widehat{\psi(\mathbf{x})} = \psi(\mathbf{y}) \oplus \text{SynDec}(\mathbf{u} \oplus \text{Syn}\,\psi(\mathbf{y})). \quad (2)$$

D. Sparse Ternary Coding (STC) [10], [17]

The encoder is a mapping $\varphi : \mathbb{R}^N \to \{-1, 0, +1\}^L$, where L may be smaller, equal to, or larger than N. The $\varphi(\mathbf{x})$ is a *sparse encoding* of $\mathbf{x}$; the number of nonzero entries is $S_t \ll L$, which is called the sparsity level. The **encoder** first
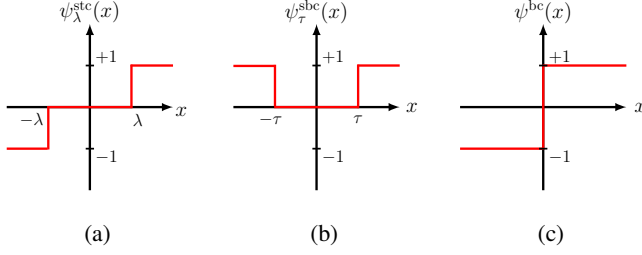
Fig. 3: *(a) Ternary thresholding; (b) Binary thresholding; (c) Binarisation.*

Fig. 4: *Enrolment phase of the SCA scheme.*

applies a projection matrix $\mathbf{W} \in \mathbb{R}^{L \times N}$ and then element-wise thresholding $\psi_\lambda^{\text{stc}}$, where λ is a parameter, $\psi_\lambda^{\text{stc}}(q) \triangleq \text{sign}(q)\Theta(|q| - \lambda)$ (see Fig. 3a).

$$\mathbf{v} = \varphi(\mathbf{x}) = \psi_\lambda^{\text{stc}}(\mathbf{W}\mathbf{x}) \in \{-1, 0, +1\}^L. \quad (3)$$

The threshold λ is tuned to get the desired sparsity level S_t. The **decoder** produces an estimator for $\mathbf{x}$ as $\widehat{\mathbf{x}} = \mathbf{W}^\dagger \mathbf{v}$.

E. Sparse Binary Coding (SBC)

Here the thresholding function is $\psi_\tau^{\text{sbc}}(x) \triangleq \Theta(|x| - \tau)$ (Fig. 3b). Given a (raw) feature vector $\mathbf{x} \in \mathbb{R}^N$ SBC generates a binary vector $\psi_\tau^{\text{sbc}}(\mathbf{x}) \in \{0, 1\}^N$.

F. Binary Coding (BC)

This is the component-wise sign operation (Fig. 3c) excluding zeros. Given a (raw) feature vector $\mathbf{x} \in \mathbb{R}^N$ the BC simply generates a binary vector $\psi^{\text{bc}}(\mathbf{x}) = \text{sign}(\mathbf{x}) \in \{-1, 1\}^N$.

G. Sparse Coding with Amibiguation (SCA) [9], [10]

Given a sparse ternary vector $\mathbf{v} = \varphi(\mathbf{x}) \in \{-1, 0, 1\}^L$, the ambiguation mechanism A turns S_n randomly chosen zero components of $\mathbf{v}$ into a (random) ± 1. The resulting ternary vector $\mathbf{u} = A(\mathbf{v})$ is stored as enrolment data (see Fig. 4), together with the matrix $\mathbf{W}$. The randomly added nonzero components make it prohibitively difficult to reconstruct $\mathbf{x}$ from $\mathbf{u}$, while still allowing a verifier to check if a verification measurement $\mathbf{y}$ is consistent with $\mathbf{u}$.

III. PROBLEM FORMULATION

We formally capture the 'medical condition' issue as follows. We model the existence of the privacy-sensitive medical condition as a binary function of one of the components of the enrollment measurement $\mathbf{x} \in \mathbb{R}^N$. That is, we say that $\psi(x_n)$ is the quantity that should not leak, for some $n \in [N]$.

We will work with two choices for the function $\psi(\cdot)$ that seem to make sense in our context: (a) $\psi(\cdot) = \psi_\tau^{\text{sbc}}(\cdot)$ (see Fig. 3b), and (b) $\psi(\cdot) = \psi^{\text{bc}}(\cdot)$ (see Fig. 3c). We will assume that the index n is not known to the legitimate parties at the time of enrollment. Otherwise, there is a trivial solution.

We will consider only distributions of X that are symmetric around $X = 0$, i.e., even functions $f(x)$. Furthermore we work in the 'perfect enrollment' model, which states that there is no measurement noise at enrollment time. In this way we are erring on the side of caution, overestimating the leakage.
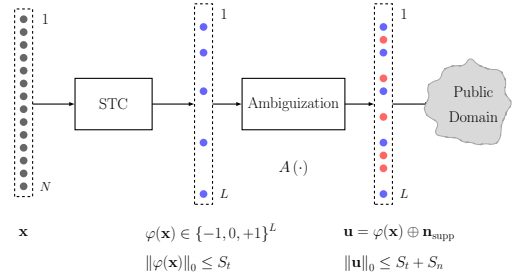
IV. RESULTS FOR HELPER DATA SYSTEMS

We zoom in on the relevant component X_n. We introduce shorthand notation $Z = \psi_\tau^{\text{sbc}}(X_n)$ and $V = \psi^{\text{bc}}(X_n)$. We write F for the cumulative distribution function of X_n.

A. Leakage from the Quantising HDS

First we look at the sign variable V.

Theorem 1. Let the distribution of X_n be an even function. Then $H(V) = 1$ and

$$H(V|U) = \begin{cases} J \text{ even} & : \quad 1, \\ J = 2t + 1, \ m \text{ even}: & h(\frac{1-p_t}{2}), \\ J = 2t + 1, \ m \text{ odd}: & \frac{m-1}{m}h(\frac{1-p_t}{2}) + \frac{1}{m}. \end{cases} \quad (4)$$

Proof. When J is even, it holds that $\Pr[V = 1] = \frac{1}{2}$ for any u. When J is odd and m is even, it holds for any u that $\Pr[V = 1|U = u] = \sum_{n>t} p_n = \frac{1}{2}\sum_{n \neq t} p_n = \frac{1}{2}(1 - p_t)$. When J is odd and m is odd, the above situation holds for $u \neq \lfloor m/2 \rfloor$, but for the one special value of u in the middle we have $\Pr[V = 1|U = \lfloor \frac{m}{2} \rfloor] = \frac{1}{2}$. □

Next we look at the threshold indicator Z. The entropy of Z is given by $H(Z) = h(F(-\tau) + 1 - F(\tau))$. For symmetric $f(x)$ this reduces to $H(Z) = h(2F(-\tau))$.

Theorem 2 below gives an expression for the entropy of Z given the helper data, in the regime where the threshold τ lies in the outermost u-region (the rightmost grayscale band in Fig. 2), i.e., the medical condition is rare.

Theorem 2. Let $f(x)$ be a symmetric pdf. Let $F(-\tau) < \frac{p_0}{m}$. Then

$$H(Z|U) = \frac{2}{m}h(mF(-\tau)). \quad (5)$$

Proof. For $u \in \{1, \ldots, m-2\}$ it is certain that $Z = 0$. This gives $H(Z|U) = \frac{1}{m}\sum_{w=0}^{m-1} H(Z|U = u) = \frac{1}{m}[H(Z|U = 0) + H(Z|U = m - 1)]$. Due to symmetry this equals $\frac{2}{m}H(Z|U = 0)$. Since Z is a binary RV we have $H(Z|U = 0) = h(\Pr[Z = 1|U = 0])$. Finally we use $\Pr[Z = 1|U = 0] = p_0 \cdot \frac{F(-\tau)}{p_0/m}$. □

Remark: For $m = 2$ we see that $H(Z|U) = H(Z)$, i.e., there is no leakage.

It is interesting to note that **in the HDS with even J and m=2, the helper data leaks absolutely nothing about V and Z.**
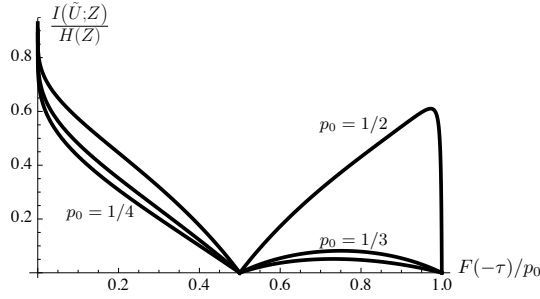
Fig. 5: *Normalised leakage $\frac{I(\tilde{U};Z)}{H(Z)}$ as a function of $F(-\tau)/p_0$. From top to bottom $p_0 = \frac{1}{2}, \frac{1}{3}, \frac{1}{4}$.*

Fig. 6: *The schematic block diagram of a physical verification system.*

One may wonder if $m > 2$ is still worth considering. It should be noted that the noise resilience of the HDS improves when the number of subdivisions m is increased. Therefore we cannot exclude that setting $m > 2$ can be a good design choice. Theorem 3 below gives a leakage result for the limiting case $m \to \infty$, i.e. the continuum helper data.

Theorem 3. Let $f(x)$ be an even function. Let $F(-\tau) \le p_0$.

$$H(Z|\tilde{U}) =$$
$$\begin{cases} F(-\tau) \le \frac{p_0}{2} : & \frac{2F(-\tau)}{p_0}h(p_0) \\ F(-\tau) > \frac{p_0}{2} : & \frac{2p_0 - 2F(-\tau)}{p_0}h(p_0) + \frac{2F(-\tau) - p_0}{p_0}h(2p_0). \end{cases} \quad (6)$$

Proof. We compute $H(Z|\tilde{U}) = \mathbb{E}_{\tilde{u}} H(Z|\tilde{U} = \tilde{u})$. Since $\tilde{u}$ is uniform on $[0,1)$ this evaluates to $\int_0^1 d\tilde{u} \, H(Z|\tilde{U} = \tilde{u}) = \int_0^1 d\tilde{u} \, h(\Pr[Z = 1|\tilde{U} = \tilde{u}])$.
Case $F(-\tau)/p_0 \le \frac{1}{2}$. For $p_0\tilde{u} \in [F(-\tau, p_0 - F(-\tau))]$ it is certain that $Z = 0$. For all other $\tilde{u}$ we have $\Pr[Z = 1|\tilde{U} = \tilde{u}] = p_0$.
Case $F(-\tau)/p_0 > \frac{1}{2}$. For $p_0\tilde{u} \in [p_0 - F(-\tau), F(-\tau)]$ we have $\Pr[Z = 1|\tilde{U} = \tilde{u}] = 2p_0$ (left tail and right tail). For all other $\tilde{u}$ the probability is p_0. $\qquad\square$

The relative leakage $[H(Z) - H(Z|\tilde{U})]/H(Z)$ is plotted in Fig. 5. The dependence on τ looks strange, but three special points can be understood. (i) For $\tau \to \infty$ the continuum helper data completely reveals Z; (ii) For $F(-\tau) = p_0$ the helper data contains no information about $|x|$ crossing the threshold. That information is contained in S, and the ZLHDS has been designed not to leak anything about S; (iii) At the intermediate value $F(-\tau)/p_0 = 1/2$ a special symmetry occurs between the left region $S = 0$ and the right region $S = J - 1$. Due to this symmetry the conditional distribution $Z|\tilde{U} = \tilde{u}$ looks the same for every value of $\tilde{u}$.
Furthermore the leakage is a decreasing function of J, since at large J the two tail regions $S = 0$ and $S = J - 1$ have less influence.

B. Leakage from the Code Offset Method

Even if the quantising HDS does not reveal V, the helper data from the Code Offset Method may still do. Typically the sign
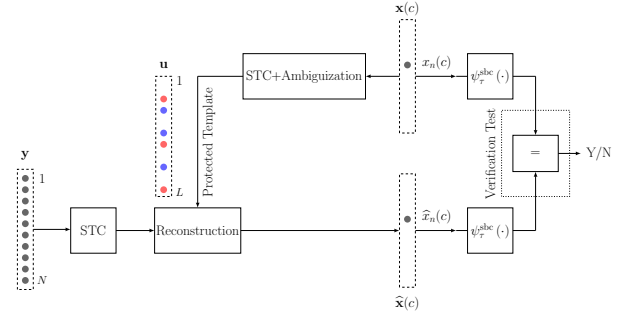
$V \in \{-1, 1\}$ gets turned into a bit value somewhere in the binary string that serves as input to the COM.
We use notation as in Section II-C. Consider *uniform* $\psi(X) \in \{0,1\}^N$, and helper data $U = \text{Syn}\,\psi(X)$. At fixed $u \in \{0,1\}^{N-k}$, there are 2^k strings x that are consistent with u, and all of them have equal probability. The marginal distribution for one component of $\psi(X)$ is uniform; hence there is no leakage about V.
The above reasoning no longer applies when $\psi(X)$ is non-uniform. Then the 2^k strings compatible with u are not uniform. However, the number of strings summed over in the computation of the marginal is exponentially large; for most 'normal-looking' $\psi(X)$-distributions the marginal of one component will still be close to uniform.
Finally we briefly depart from the *perfect enrollment* setting and investigate the effect of measurement noise at enrollment time on the privacy properties of the COM. Suppose there exists a 'true' biometric $B \in \{0,1\}^N$, which is shielded from our view by enrollment noise G. The enrollment measurement yields $B \oplus G$. Then the relevant privacy question is how much is leaking about (parts of) B, as opposed to $B \oplus G$.

Proposition 1. Let the enrollment noise G be bitwise iid Bernoulli noise with bit error rate ε. Let r be the row weight of the error correcting code. Let $U = \text{Syn}(B \oplus G)$. Then

$$I(B; U) \approx (N - k)[1 - h\left(\frac{1}{2} - \frac{1}{2}(1 - 2\varepsilon)^r\right)]. \quad (7)$$

Proof. $I(B; U) = H(U) - H(U|B) = H(U) - H(\text{Syn}\,B \oplus \text{Syn}\,G|B) = H(U) - H(\text{Syn}\,G)$. We use $H(U) \le N - k$. In a good code the redundancy $N - k$ is just slightly larger than the entropy of the syndrome $\text{Syn}\,G$. We estimate the entropy of $\text{Syn}\,G$ by setting it close to the Gallager bound [18], $H(\text{Syn}\,G) \approx (N - k)h(\alpha)$, where α is the bit error probability of r binary symmetric channels concatenated together, $\alpha = \frac{1}{2} - \frac{1}{2}(1 - 2\varepsilon)^r$. $\qquad\square$

For reasonable values of the row weight, we see that the leakage $I(B; U)$ is very small, approximately $(N-k)\frac{(1-2\varepsilon)^{2r}}{2\ln 2}$; and this is leakage about the whole vector B.
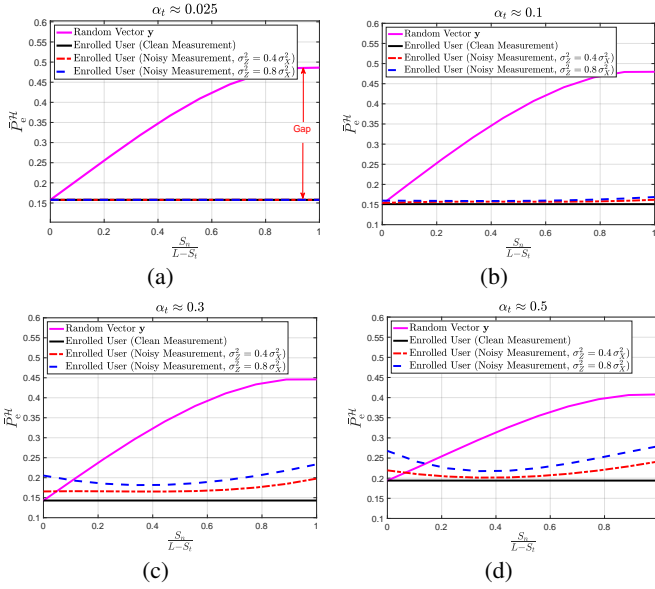
Fig. 7: *Error probabilities for different sparsity ratios α_t and measurement noise σ_Z^2; setting $\sigma_X^2 = 0.5$, $\tau = \sigma_X$.*

V. RESULTS FOR SPARSE CODING WITH AMBIGUATION

A. Method

We consider C users. Each user $c \in [C]$ has a measurement vector $\mathbf{x}(c)$; it consists of N components which are modeled as zero-mean unit-variance Gaussian variables. The enrolment of the vectors $\mathbf{x}(1), \ldots, \mathbf{x}(C)$ is as described in Section II-G. The public data for user c is $\mathbf{W}(c), \mathbf{u}(c)$.

We consider a verification vector $\mathbf{y} \in \mathbb{R}^N$, which is allegedly from user c. It is either a noisy version of the enrolled $\mathbf{x}(c)$ or completely unrelated to it (but drawn from the same distribution). The former case is referred to as Hypothesis $\mathcal{H}_1$, the latter as $\mathcal{H}_0$. We write $\mathbf{y}^{\text{auth}}(c) = \mathbf{x}(c) + \mathbf{z}$, with Gaussian noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma_Z^2 \mathbf{I})$. The verification procedure works as follows. The verifier computes $\mathbf{u}(c) \cdot \psi_\lambda^{\text{stc}}(\mathbf{W}(c)\mathbf{y})$. If this inner product exceeds a threshold, then he decides on $\mathcal{H}_1$, otherwise $\mathcal{H}_0$. The expression $\psi_\lambda^{\text{stc}}(\mathbf{W}(c)\mathbf{y})$ is essentially an STC 'enrollment' of $\mathbf{y}$ without ambiguation noise. Taking the inner product with $\mathbf{u}(c)$ is meant to remove the ambiguation noise from $\mathbf{u}(c)$ (we call this 'purification'), and it results in a similarity score.

The goal of the SCA mechanism is to prevent recovery of $\mathbf{x}$ and the corresponding $\psi(x_n)$ from the enrolment data while enabling a verifier to check if $\mathbf{y}$ is consistent with enrolment data $\mathbf{u}(c)$. We can characterize the performance of SCA in terms of:

(i) preservation of mutual information between $\mathbf{X}$ and $\widehat{\mathbf{X}}$ in the authorized case, i.e. $I(\mathbf{X}; \widehat{\mathbf{X}}|\mathcal{H}_1) \to H(\mathbf{X})$, whilst in the unauthorized case $I(\mathbf{X}; \widehat{\mathbf{X}}|\mathcal{H}_0) \to 0$. The same holds for a function of $\mathbf{X}$, i.e., $I(\psi(X_n); \psi(\widehat{X}_n)|\mathcal{H}_1) \to H(\psi(X_n))$ and $I(\psi(X_n); \psi(\widehat{X}_n)|\mathcal{H}_0) \to 0$.

(ii) Reconstruction error: We investigate the error probabilities

$$P_{\text{e}}^{\mathcal{H}}(c) = \Pr[\psi_\tau^{\text{sbc}}(\widehat{X}_n(c)) \neq \psi_\tau^{\text{sbc}}(X_n(c))|\mathcal{H}]. \quad (8)$$
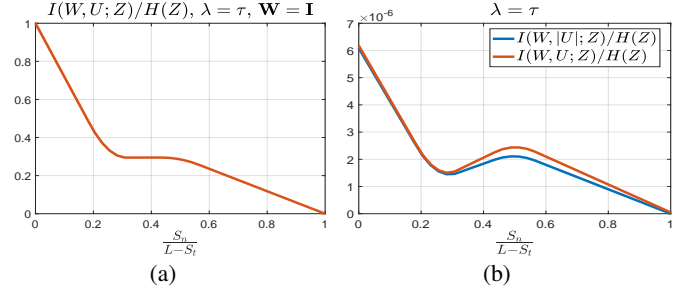

Fig. 8: *Normalized Information Leakage $\frac{I(WU;Z)}{H(Z)}$ as a function of ambiguation ratio $\frac{S_n}{L-S_t}$. $\lambda = \tau = \sigma_X$. (a) $\mathbf{W} = \mathbf{I}$; (b) $\mathbf{W}$ equals the PCA transform of the matrix $[\mathbf{x}(1) \cdots \mathbf{x}(C)]$.*

Let $p \triangleq \Pr[\psi_\tau^{\text{sbc}}(X_n) = 1]$, $p < \frac{1}{2}$. Ideally it should hold that $P_{\text{e}}^{\mathcal{H}_1} \to 0$ and $P_{\text{e}}^{\mathcal{H}_0} \to 2p(1-p)$. The latter expression is for random $\widehat{X}_n$ independent of X_n, with the same distribution.

B. Performance results

We consider a database $\mathbf{X}$ of $C = 100,000$ random vectors (individuals) with dimensionality $N = 256$, which are generated from the distribution $\mathcal{N}(\mathbf{0}, \sigma_X^2 \mathbf{I})$. We then generate the noisy version of $\mathbf{X}$ with two different noise variances $\sigma_Z^2 = 0.4\sigma_X^2$ and $\sigma_Z^2 = 0.8\sigma_X^2$. We consider square matrix $\mathbf{W}$, i.e., $L = N$.

We look at the error probability averaged over the users and the components, $\overline{P}_{\text{e}}^{\mathcal{H}} = \frac{1}{CN} \sum_{n=1}^{N} \sum_{c=1}^{C} P_{\text{e}}^{\mathcal{H}}(c)$.

Fig. 7 shows the averaged error probability $\overline{P}_e^{\mathcal{H}}$ as a function of the ambiguation ratio $\frac{S_n}{L-S_t}$, for fixed $\sigma_X^2 = 0.5$, $\tau = \sigma_X$ and different sparsity ratios $\alpha_t \triangleq S_t/L = 0.025, 0.1, 0.3, 0.5$ and measurement noise variances $\sigma_Z^2 = 0.4\sigma_X^2, 0.8\sigma_X^2$. Several things are worth noting.

- For the un-enrolled case (random $\mathbf{y}$), the error probability in guessing the privacy bit Z increases as a function of the ambiguation ratio. This is as expected.
- In the genuine user case the situation is more complex; the ambiguation noise interferes with the measurement noise.
- There is a clear gap between the genuine user case and the un-enrolled case. The low[1] (and in some plots nearly constant) error rate for genuine users demonstrates that the 'purification' correctly removes the ambiguation noise.

Furthermore we compute the leakage $I(\mathbf{W}, \mathbf{U}; Z)/H(Z)$. Fig. 8a shows what happens when $\mathbf{W}$ is set to the trivial value $\mathbf{W} = \mathbf{I}$. The leakage decreases from 100% to zero with increasing ambiguation. The curve seems to consist of three ambiguation ratio regimes, with piecewise linear behaviour: $0 - 0.25$, $0.25 - 05$ and $0.5 - 1$. At the moment we are not able to explain this behaviour.

Fig. 8b shows what happens when a less trivial matrix $\mathbf{W}$ is used, namely the PCA transform matrix of the matrix

---

[1]Note that the False Negative probability for the overall user matching is much lower than the single-component reconstruction error.

$[\mathbf{x}(1) \cdots \mathbf{x}(C)]$. This same $\mathbf{W}$ is used for all users. We again observe piecewise linear behaviour with the same three intervals. However, the middle piece is no longer constant but increasing. More importantly, the leakage is reduced by orders of magnitude. Fig. 8b also shows the leakage from Sparse Binary Coding with ambiguation; it is slightly smaller than for the ternary case.

C. Non-square projection matrix

We briefly discuss the case $L > N$, i.e. the number of random projections is larger than the dimension of $\mathbf{x}$. The adversary is confronted with an ambiguized ternary vector $\mathbf{u}$, and from it has to guess the $\mathbf{v} = \psi_\lambda^{\text{stc}}(\mathbf{W}\mathbf{x})$ by guessing which locations in $\mathbf{u}$ contain the ambiguation noise. When L is larger than N, the adversary may be able to distinguish between wrong guesses and the correct guess, as follows. For a wrong guess it will typically hold that $WW^\dagger \mathbf{v}_{\text{wrong}}$ is far away from $\mathbf{v}_{\text{wrong}}$, while on the other hand it holds that $WW^\dagger \mathbf{v} \approx \mathbf{v}$. From the correct $\mathbf{v}$ an estimator for $\mathbf{x}$ is then obtained as $\widehat{\mathbf{x}} = W^\dagger \mathbf{v}$. Hence, information-theoretically speaking, there is no privacy protection. However, the amount of effort in going through all the possible guesses scales as $\binom{S_t + S_n}{S_n}$, which is huge. The security is *computational*, not information-theoretic.

VI. Discussion

For quantizing HDSs we have established that there is a clearly identifiable optimal choice for protecting the V and Z bits: taking the number of quantization intervals to be even, and setting $m = 2$. However, for noise tolerance it is advantageous to set m as large as possible. The Z-leakage result for $m \to \infty$ (Fig. 5) has some caveats. It is nice that a minimum exists at $F(-\tau) = p_0/2$ and $F(-\tau) = p_0$, but unfortunately the operational meaning of τ is not really well defined. A small shift of τ has little impact on the concept "this variable is abnormally far from zero", but has a large effect in Fig. 5. It is left as a topic for future work to study this further.

There are no such subtleties for the Code Offset Method. We think we can safely conclude that the COM has only negligible leakage.

For the SCA approach we have established that there is a clear gap between how much you know about Z if you *do* and *do not* have access to a matching verification measurement $\mathbf{y}(c)$. (Not having such access means trying to reconstruct Z from the public data.) This is visible as a gap (Fig. 7) in the error probability for reconstructing Z, and as low mutual information $I(\mathbf{W}, \mathbf{U}; Z)$ in Fig. 8b. Determining the leakage about $\text{sign}(X_n)$ is left for future work. Other topics for future work are further experimentation with different choices of the projection matrix $\mathbf{W}$ and understanding the piecewise linear shape of the leakage curve.

References

[1] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM Conference on Computer and Communications Security (CCS) 1999*, 1999, pp. 28–36.

[2] J.-P. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *Audio- and Video-Based Biometric Person Authentication*. Springer, 2003.

[3] Y. Dodis, M. Reyzin, and A. Smith, "Fuzzy Extractors: How to generate strong keys from biometrics and other noisy data," in *Eurocrypt 2004*, ser. LNCS, vol. 3027. Springer-Verlag, 2004, pp. 523–540.

[4] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy Extractors: how to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[5] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[6] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.

[7] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, 2012.

[8] C. Aguilar-Melchor, S. Fau, C. Fontaine, G. Gogniat, and R. Sirdey, "Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 108–117, 2013.

[9] 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.

[10] B. Razeghi and S. Voloshynovskiy, "Privacy-preserving outsourced media search using secure sparse ternary codes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, April 2018, pp. 1–5.

[11] B. Razeghi, S. Voloshynovskiy, S. Ferdowsi, and D. Kostadinov, "Privacy-preserving identification via layered sparse code design: Distributed servers and multiple access authorization," in *26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018.

[12] S. Rezaeifar, B. Razeghi, O. Taran, T. Holotyak, and S. Voloshynovskiy, "Reconstruction of privacy-sensitive data from protected templates," in *IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, September 2019.

[13] E. Verbitskiy, P. Tuyls, C. Obi, B. Schoenmakers, and B. Škorić, "Key extraction from general nondiscrete signals," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 269–279, 2010.

[14] J. de Groot, B. Škorić, N. de Vreede, and J. Linnartz, "Quantization in Zero Leakage Helper Data Schemes," *EURASIP Journal on Advances in Signal Processing*, 2016, 2016:54.

[15] T. Stanko, F. Andini, and B. Škorić, "Optimized quantization in Zero Leakage Helper Data Systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1957–1966, 2017.

[16] C. Bennett, G. Brassard, C. Crépeau, and M. Skubiszewska, "Practical quantum oblivious transfer," in *CRYPTO*, 1991, pp. 351–366.

[17] S. Ferdowsi, S. Voloshynovskiy, D. Kostadinov, and T. Holotyak, "Sparse ternary codes for similarity search have higher coding gain than dense binary codes," in *IEEE Int. Symp. on Inf. Theory (ISIT), 2017*.

[18] R. Gallager, *Low Density Parity Check Codes*. MIT Press, 1963.