AGGREGATION AND EMBEDDING FOR GROUP MEMBERSHIP VERIFICATION

Marzieh Gheisari†, Teddy Furon†, Laurent Amsaleg†, Behrooz Razeghi⋆, Slava Voloshynovskiy⋆

† Univ Rennes, Inria, CNRS, IRISA, France, ⋆University of Geneva, Switzerland

{marzieh.gheisari-khorasgani, teddy.furon}@inria.fr, laurent.amsaleg@irisa.fr,

{behrooz.razeghi, svolos}@unige.ch

ABSTRACT

This paper proposes a group membership verification protocol preventing the curious but honest server from reconstructing the enrolled signatures and inferring the identity of querying clients. The protocol quantizes the signatures into discrete embeddings, making reconstruction difficult. It also aggregates multiple embeddings into representative values, impeding identification. Theoretical and experimental results show the trade-off between the security and the error rates.

***Index Terms*—** group representation, discrete embedding, aggregation, data privacy, verification.

1. INTRODUCTION

Verifiying that an item/device/individual is a member of a group is needed for many applications granting or refusing access to sensitive resources. Group membership verification is *not* about identifying first and then checking membership. Rather, being granted with access requires that the members of the group could be distinguished from non-members, but it does not require to distinguish members from one another.

Group membership verification protocols first enroll eligible *signatures* into a data structure stored at a server. Then, at verification time, the structure is queried by a client signature and the access is granted or not. For security, the data structure must be adequately protected so that a honest but curious server cannot reconstruct the signatures. For privacy, verification should proceed anonymously, not disclosing identities.

A client signature is a noisy version of the enrolled one, *e.g.* due to changes in lighting conditions. The verification must absorb such variations and cope with the continuous nature of signatures. They must be such that it is unlikely that a noisy version for one user gets similar enough to the enrolled signature of any other user. Continuity, discriminability and statistical independence are inherent properties of signatures.

This paper proposes a group membership verification protocol preventing a curious but honest server from reconstructing the enrolled signatures and inferring the identity of querying (trusted) clients. It combines two building blocks:

Block #1: One building block hashes continuous vectors into discrete *embeddings*. This lossy process limits the ability of the server to reconstruct signatures from the embeddings.

Research supported by the ERA-Net project ID_IoT 20CH21_167534.

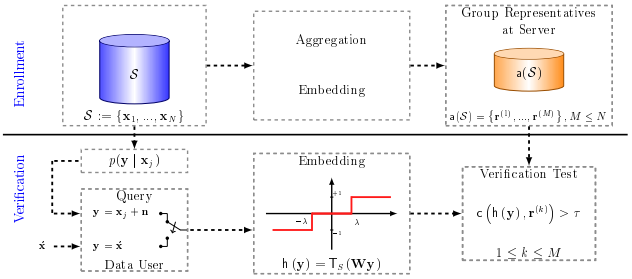

Fig. 1: Block diagram of the proposed model.

Block #2: The other building block *aggregates* multiple vectors into a unique representative value which will be enrolled at the server. The server can therefore not infer any specific signature from this value. Sufficient information must be preserved through the aggregation process for the server to assert whether or not a querying signature is a member of the group.

These two blocks can be assembled according to two configurations: block #1 before block #2, the system acquires and then hashes the signatures before aggregating them. The opposite configuration is where acquired signatures are aggregated before hashing the result of this aggregation. At query time, the newly acquired signature is always hashed before being sent to the server. Weaknesses and strengths of these two configurations are explored in the paper.

2. RELATED WORK

Group membership verification protocols relying on cryptography exist [1] but are more relevant to authentication, identification and secret binding. Other approaches apply homomorphic encryption to signatures, compare [2] and threshold them [3, 4] in the encrypted domain, and need active participation of clients. Approaches involving cryptography, however, are extremely costly, memory and CPU wise.

Group membership is linked to Bloom filters that are used to test whether an element is a member of a set. When considering security, a server using Bloom filters cannot infer any information on one specific entry [5]. Note that Bloom filters can not deal with continuous high dimensional signatures and that queries must be encrypted to protect the privacy of users [6, 7]. Bloom filters, adapted to our setup, however,

form a baseline in our experiments (see Sect. 5.3).

Embedding *a single* high dimensional signature is quite a standard technique. The closest to our work is the privacy-preserving identification mechanism based on sparsifying transform [8, 9, 10]. It produces an information-preserving sparse ternary embedding, ensuring privacy of the data users and security of the signature.

Aggregating signals into similarity-preserving representations is very common in computer vision [11, 12, 13]. They do not consider security or privacy. In [14], Iscen *et al.* use the *group testing* paradigm to pack a random set of image signatures into a unique high-dimensional vector. It is therefore an excellent basis for the aggregation block: the similarities between the original non-aggregated signatures and a query signature is preserved through the aggregation.

3. NOTATIONS AND DEFINITIONS

Signatures are vectors in $\mathbb{R}^d$. If N users/items belong to the group, then the protocol considers N signatures, $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$. The signature to verify is a query vector $\mathbf{y} \in \mathbb{R}^d$. Group membership verification considers two hypotheses linked to the continuous nature of the signatures:

$\mathcal{H}_1$: The query is related to one of the N vectors. For instance, it is a noisy version of vector j, $\mathbf{y} = \mathbf{x}_j + \mathbf{n}$, with $\mathbf{n}$ to be a noise vector.

$\mathcal{H}_0$: The query is not related to any vector in the group.

We first design a group aggregation technique s which computes a single representation from all N vectors $\mathbf{r} := \mathsf{s}(\mathcal{S})$. This is done at the enrollment phase. Variable ℓ denotes the size in bits of this representation.

At the verification phase, the query $\mathbf{y}$ is hashed by a function h of size ℓ in bits. This function might be probabilistic to ensure privacy. The group membership test decides which hypothesis is deemed true by comparing $\mathsf{h}(\mathbf{y})$ and $\mathbf{r}$. This is done by first computing a score function c and thresholding its results: $t := [\mathsf{c}(\mathsf{h}(\mathbf{y}), \mathbf{r}) > \tau]$.

3.1. Verification Performances

The performances of this test are measured by the probabilities of false negative, $p_{\mathsf{fn}}(\tau) := \mathbb{P}(t = 0|\mathcal{H}_1)$, and false positive, $p_{\mathsf{fp}}(\tau) := \mathbb{P}(t = 1|\mathcal{H}_0)$. As τ varies from $-\infty$ and $+\infty$, these measures are summarized by the AUC (Area Under Curve) performance score. Another figure of merit is $p_{\mathsf{fn}}(\tau)$ for τ s.t. $p_{\mathsf{fp}}(\tau) = \epsilon$, a required false positive level.

3.2. Security and Privacy

A curious server can reconstruct a signature $\mathbf{x}$ from its embedding (for instance the query): $\hat{\mathbf{x}} = \mathsf{rec}(\mathsf{h}(\mathbf{x}))$. The mean squared error is a way to assess its accuracy: $\mathsf{MSE} = \mathbb{E}(\|\mathbf{X} - \mathsf{rec}(\mathsf{h}(\mathbf{X}))\|^2)/d$. The best reconstruction is known to be the conditional expectation: $\hat{\mathbf{x}} = \mathbb{E}(\mathbf{X}|\mathsf{h}(\mathbf{x}))$.

Reconstructing an enrolled signature from the group representation is even more challenging. Due to the *aggregation* block, the curious server can only reconstruct a single vector

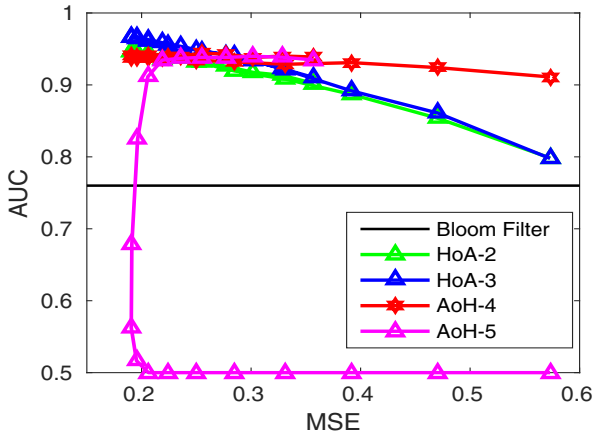

Fig. 2: Unique group: AUC vs. $\mathsf{MSE}_q/\sigma_y^2$. $N = 128$, $d = 1024$, $\sigma_n^2 = 0.01$ for varying $S \in (0.1 \times d, 0.9 \times d)$.

$\hat{\mathbf{x}}$ from the aggregated representation, and this vector serves as an estimation of any signatures in the group:

$$\mathsf{MSE}_e = (dN)^{-1} \sum_{j=1}^{N} \mathbb{E}(\|\mathbf{X}_j - \hat{\mathbf{X}}\|^2). \quad (1)$$

4. VERIFICATION FOR A FEW GROUP MEMBERS

This section discusses the verification protocol when N is small. We study the two different configurations for assembling block #1 and block #2.

Block #1: Embedding. An embedding $\mathsf{h} : \mathbb{R}^d \to \mathcal{A}^\ell$ maps a vector to a sequence of ℓ discrete symbols. This quantization shall preserve enough information to tell whether two embeddings are related, but not enough to reconstruct a signature. We use the sparsifying transform coding [8, 9]. It projects $\mathbf{x} \in \mathbb{R}^d$ to the range space of a transform matrix $\mathbf{W} \in \mathbb{R}^{\ell \times d}$. The output alphabet $\mathcal{A} = \{-1, 0, +1\}$ is imposed by quantizing the components of $\mathbf{Wx}$ whose amplitude is lower than λ to 0, the others to +1 or -1 according to their sign. In expectation, $S = 2d\Phi(-\lambda/\sigma_x)$ symbols are non null.

Block #2: Aggregation. Aggregation a processes a set of input vectors to produce a unique output vector. When block

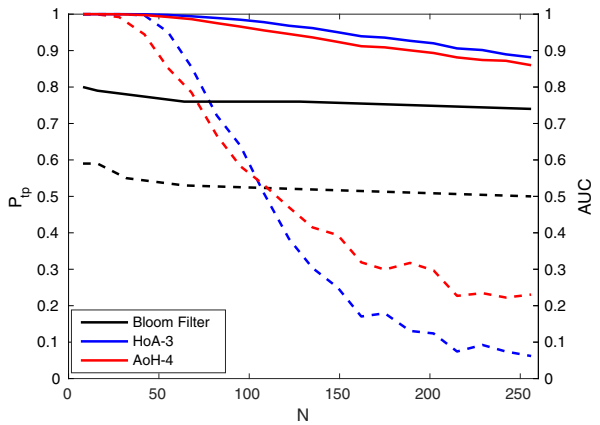

Fig. 3: Unique group: AUC and p_{tp} vs. N. Solid and dashed lines correspond to AUC and $p_{\mathsf{tp}}@p_{\mathsf{fp}} = 10^{-2}$.

#1 is used before, that is, when considering $\mathsf{s} = \mathsf{a} \circ \mathsf{h}$, then $\mathsf{a} : \mathbb{R}^{\ell \times N} \to \mathbb{R}^{\ell}$. When block #2 is used before block #1, that is when considering $\mathsf{s} = \mathsf{h} \circ \mathsf{a}$, then $\mathsf{a} : \mathbb{R}^{d \times N} \to \mathbb{R}^{d}$.

4.1. Aggregation strategies

The nature of a highly depends on the type of vector the aggregation function receives. When considering $\mathsf{s} = \mathsf{h} \circ \mathsf{a}$, then a gets continuous signatures. In this case it is possible to design two aggregations schemes that are:

$$\mathsf{a}(\mathcal{S}) = \sum_{\mathbf{x} \in \mathcal{S}} \mathbf{x} = \mathbf{G}\mathbf{1}_N \quad \text{or} \tag{2}$$

$$\mathsf{a}(\mathcal{S}) = (\mathbf{G}^{\dagger})^{\top}\mathbf{1}_N. \tag{3}$$

where $\mathbf{G}$ is the $d \times N$ matrix $\mathbf{G} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, $\mathbf{1}_N := (1, \ldots, 1)^{\top} \in \mathbb{R}^N$, and $\mathbf{G}^{\dagger}$ is the pseudo-inverse of $\mathbf{G}$. Eq. (2) is called the sum and (3) the pinv schemes in [14].

When considering $\mathsf{s} = \mathsf{a} \circ \mathsf{h}$, then a gets the embeddings of the signatures. Two additional aggregation strategies are the sum and sign pooling (4) and the majority vote (5):

$$\mathbf{r} = \mathsf{sign}(\sum_{\mathbf{x} \in \mathcal{S}} \mathsf{h}(\mathbf{x})) \quad \text{or} \tag{4}$$

$$r_i = \arg \max_{s \in \{-1,0,1\}} |\{\mathbf{x} \in \mathcal{S} | \mathsf{h}(\mathbf{x})_i = s\}| \tag{5}$$

4.2. Four resulting schemes

The assemblage of the blocks and the aggregation strategies overall create four variants. We name them:

- **HoA-2:** this scheme sums the raw signatures into a unique vector before embedding it in order to obtain $\mathbf{r}$. It therefore corresponds to the case where $\mathsf{s} = \mathsf{h} \circ \mathsf{a}$, the aggregation a being defined by (2).
- **HoA-3:** here also, aggregation precedes embedding, $\mathsf{s} = \mathsf{h} \circ \mathsf{a}$, and a is defined by (3).
- **AoH-4:** this scheme embeds each signature before aggregating with sum and sign pooling as defined by (4).
- **AoH-5:** here also, embedding precedes aggregation, but the majority vote is used as defined by (5).

The score function c comparing the hashed query with the group representation is always $\mathsf{c}(\mathsf{h}(\mathbf{y}), \mathbf{r}) = -\|\mathsf{h}(\mathbf{y}) - \mathbf{r}\|$.

5. RECONSTRUCTION AND VERIFICATION

This section makes the following assumptions: i) Enrolled signatures are modelled by $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_d, \sigma_x^2 \mathbf{I}_d)$, ii) Square orthogonal matrix $\mathbf{W}$ known by the attacker.

5.1. Ability to reconstruct from the embedding

Now that $\mathbf{W}$ preserves the norm, the MSE on $\mathbf{X}$ is the same as the mean square reconstruction error on $\mathbf{Z} = \mathbf{W}\mathbf{X}$, which is also white Gaussian distributed. Thanks to the independance of the components of $\mathbf{Z}$, the conditional expectation can be computed component-wise. We introduce the density function conditioned on the interval $\mathcal{R}_s \subset \mathbb{R}$:

$$f(z|\mathcal{R}_s) := \phi_{\sigma_x}(z).\mathbb{1}_{\mathcal{R}_s}(z)/\mathbb{P}(Z \in \mathcal{R}_s), \tag{6}$$

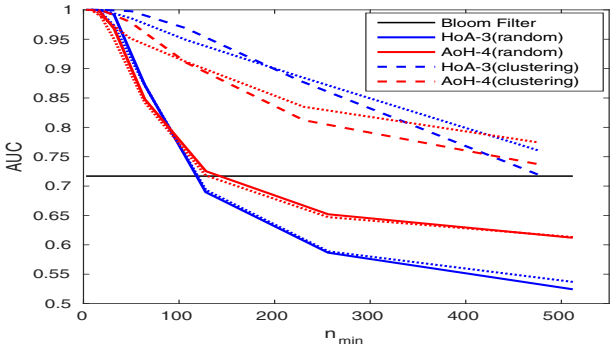

Fig. 4: Multiple groups: AUC vs. $n_{\min}$. Dotted lines are theoretical AUC. $N = 4096$, $d = 1024$, $\sigma_n^2 = 10^{-2}$, $S/d = 0.6$ for HoA-6, and $S/d = 0.85$ for AoH-7.

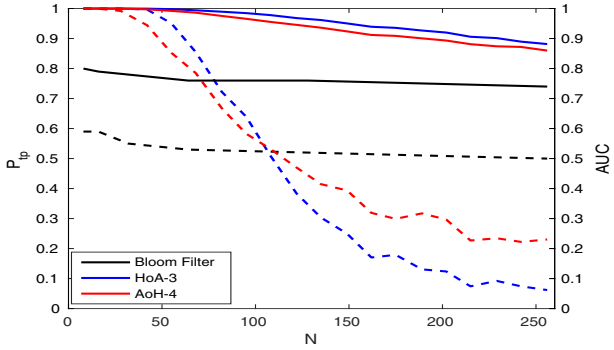

Fig. 5: Multiple groups: MSE_e vs. n_{min}. $N = 4096$, $d = 1024$, $\sigma_n^2 = 10^{-2}$, $S/d = 0.6$ (HoA-6) or 0.85 (AoH-7).

with intervals $\mathcal{R}_0 = [-\lambda, \lambda]$, $\mathcal{R}_1 = (\lambda, +\infty)$, and $\mathcal{R}_{-1} = (-\infty, -\lambda)$. Function ϕ_{σ_x} is the p.d.f. of $Z \sim \mathcal{N}(0; \sigma_x^2)$ and $\mathbb{1}_{\mathcal{R}_s}$ is the indicator function of interval $\mathcal{R}_s$.

Observing the i-th symbol of $\mathsf{h}(\mathbf{x})$ equals s reveals that $z_i \in \mathcal{R}_s$. This component is reconstructed as $\hat{z}_i(s) := \mathbb{E}(Z|\mathcal{R}_s)$. Note that $\hat{z}_i(0) = 0$ because $f(z|\mathcal{R}_0)$ is symmetric around 0. For $s = 1$, the reconstruction value equals $\hat{z}_i(1) = \int_{-\infty}^{+\infty} z.f(z|\mathcal{R}_1)dz = \frac{\sigma_y}{p_1\sqrt{2\pi}} e^{-\frac{\lambda^2}{2\sigma_x^2}}$, where $p_1 := \mathbb{P}(Z \in \mathcal{R}_1) = \Phi(-\lambda/\sigma_x)$. By symmetry, $\hat{z}_i(-1) = -\hat{z}_i(1)$, and MSE admits the following close form:

$$\mathsf{MSE} = \sigma_x^2.\mathsf{MSE}(\lambda) \tag{7}$$

$$\mathsf{MSE}(\lambda) := 1 - \frac{1}{\pi\Phi(-\lambda/\sigma_x)} e^{-\frac{\lambda^2}{\sigma_x^2}}. \tag{8}$$

This quantity starts at $1 - 2\pi^{-1}$ when $\lambda = 0$. The embeddings are then full binary words ($p_1 = 1/2$). All components are reconstructed by $\pm\hat{z}_i$ but with a large variance. As λ increases, this variance decreases but less non-null components are reconstructed. $\mathsf{MSE}(\lambda)$ achieves a minimum of ≈ 0.19 for $\lambda \approx 0.60$, where 55% of the symbols of an embedding are non null. Then, $\mathsf{MSE}(\lambda)$ increases up to 1 for a large λ: the embeddings becomes sparser and sparser. When fully zero, each component is reconstructed by 0, and MSE equals σ_x^2.

5.2. Ability to reconstruct the signatures

The curious server tries to reconstruct a unique vector $\hat{\mathbf{x}}$ from $\mathbf{r}$ which represents the N enrolled signatures. Note that $\mathbf{r}$ is scale invariant: scaling the signatures by any positive factor does not change $\mathbf{r}$. Suppose that the curious server reconstructs $\hat{\mathbf{x}} = \kappa\mathbf{u}$. The best scaling minimizing MSE_e (1) is: $\kappa^\star = \|\mathbf{u}\|^{-2}\mathbf{u}^\top\mathbf{m}$, with $\mathbf{m} := N^{-1}\sum_{j=1}^N \mathbf{x}_j$. The curious server can not compute $\kappa^\star$ giving birth to a larger distortion:

$$\mathsf{MSE}_e \geq \sum_{j=1}^{N} \|\mathbf{x}_j\|^2 - N\frac{(\mathbf{u}^\top\mathbf{m})^2}{\|\mathbf{u}\|^2}. \tag{9}$$

This lower bound is further minimized by choosing $\mathbf{u} \propto \mathbf{m}$.

Therefore, aggregation (2) is less secure as the other schemes do not allow the reconstruction of $\mathbf{m}$. In the worst case (2), the curious server estimates $\mathbf{m}$ by $N^{-1}\mathsf{rec}(\mathsf{h}(\mathsf{a}(\mathcal{S})))$:

$$\begin{aligned} d.\mathsf{MSE}_e &= \mathbb{E}\|\mathbf{X}_j - N^{-1}\mathsf{rec}(\mathsf{a}(\mathcal{S}))\|^2 \qquad (10)\\ &= \mathbb{E}\|\mathbf{X}_j - \frac{\mathsf{a}(\mathcal{S})}{N}\|^2 + \frac{\mathbb{E}\|\mathsf{a}(\mathcal{S}) - \mathsf{rec}(\mathsf{a}(\mathcal{S}))\|^2}{N^2}. \end{aligned}$$

The first term is the squared distance between $\mathbf{X}_j$ and $\mathbf{m}$, whereas the second term corresponds to the error reconstruction for inverting the embedding. In the end:

$$\mathsf{MSE}_e = \sigma_x^2\left(1 - \frac{1}{N}(1 - \mathsf{MSE}(\lambda))\right). \tag{11}$$

This figure of merit increases with N because $\mathsf{MSE}(\lambda) \leq 1$, $\forall\lambda \geq 0$: Packing more signatures increases security.

5.3. Verification performances

We compare to a baseline defined as a Bloom filter optimally tuned for given N and p_{fp} having length $\ell_B = \lceil N|\log p_{\mathsf{fp}}|\log(2)^{-2}\rceil$. An embedding h is mandatory to first turn the real signatures into discrete objects. This means that, under $\mathcal{H}_1$, a false negative happens whenever $\mathsf{h}(\mathbf{x}_j + \mathbf{n}) \neq \mathsf{h}(\mathbf{x}_j)$.

Fig. 2 shows the AUC vs. MSE (7) for the schemes of Sect. 4.1 for different sparsity S/d. Two schemes performs better. For low privacy (small MSE_q), HoA-3 achieves the largest AUC (with $0.5 \leq S/d \leq 0.6$) ; for high privacy, AoH-4 is recommended (with $S/d \geq 0.85$). In these regimes, the performances are better than the Bloom filter.

Fig. 3 shows how the verification performances decrease as the number N of enrolled signatures increases. As mentioned in [14], the behavior of the aggregation scheme depends on the ratio N/d. The longer the signatures, the more of them can be packed into one representation.

6. VERIFICATION FOR MULTIPLE GROUPS

When N is large, aggregating all the signatures into a unique $\mathbf{r}$ performs poorly. Rather, for large N, we propose to partition the enrolled signature into $M > 1$ groups, and to compute M different representatives, one per partition.

Random assignment: The signatures are randomly assigned into M groups of size $n = N/M$.

Clustering: Similar signatures are assigned to the same group. The paper uses the k-means algorithm to do so. Yet, the size of the groups is no longer constant.

6.1. Verification performances

Denote by $(p_{\mathsf{fp}}^{(k)}, p_{\mathsf{tp}}^{(k)})$ the operating point of group number k, $1 \leq k \leq M$. The overall system outputs a positive answer when at least one group test is positive. Denote by $(P_{\mathsf{fp}}(M), P_{\mathsf{tp}}(M))$ the performance of the global system. Under $\mathcal{H}_0$, the query is not related to any vector. Therefore,

$$P_{\mathsf{fp}}(M) = 1 - \prod_{k=1}^{M}(1 - p_{\mathsf{fp}}^{(k)}), \tag{12}$$

Under $\mathcal{H}_1$, the query is related to only one vector belonging to one group. A false negative occurs, if this test produces a false negative and the other tests a true negative each:

$$P_{\mathsf{fn}}(M) = \sum_{k=1}^{M}\frac{n_k}{N}p_{\mathsf{fn}}^{(k)}\prod_{l\neq k}(1 - p_{\mathsf{fp}}^{(l)}). \tag{13}$$

The operating point of a group test is mainly due to the size of the group. The random assignment creates even groups (if M divides N), so these share the operating point $(p_{\mathsf{fp}}, p_{\mathsf{tp}})$.

Fig. 4 shows the experimental AUC and the one predicted by (12) and (13) when M ranges from 8 to 512. Since clustering makes groups of different sizes, we show the performances versus $n_{min} = \min_{1\leq k\leq M}(n_k)$, where n_k is the size of k-th group. The theoretical formulas are more accurate for random partitioning where the group are even. Estimations of $(p_{\mathsf{fp}}^{(k)}, p_{\mathsf{fn}}^{(k)})$ were less precise with the clustering strategy, and this inaccuracy cumulates in (12) and (13).

Clustering improves the verification performances a lot especially for HoA-3. A similar phenomenon was observed in [14]. Yet, Fig. 5 shows that it does not endanger the system: MSE_e is only slightly smaller than for random assignment, and indeed close to 1 for $n_{\min} \geq 100$. This is obtained for $M = 32$ for HoA-3 giving $\mathsf{AUC} = 0.97$. The space is so big that the clusters are gigantic and not revealing much about where the signatures are. However, the anonymity is reduced because the server learns which group provided a positive test. This is measured in term of k-anonymity by the size of the smallest group, *i.e.* $n_{\min}$. Fig. 4 indeed shows the trade-off between k-anonymity and the verification performances.

7. CONCLUSION

This paper proposed four schemes for verifying the group membership of continuous high dimensional vectors. The keystones are the aggregation and embedding functions. They prevent accurate reconstruction of the enrolled signatures, while recognizing noisy version. However, the anonymity is slightly revealed when managing many signatures aggregated into several representatives: the server is only able to link each signature to its group number. Yet, the full identity of the user is preserved.

8. REFERENCES

[1] Stuart Schechter, Todd Parnell, and Alexander Hartemink, "Anonymous authentication of membership in dynamic groups," in *Proceedings of the International Conference on Financial Cryptography*, 1999.

[2] Juan Ramón Troncoso-Pastoriza, Daniel González-Jiménez, and Fernando Pérez-González, "Fully private noninteractive face verification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1101–1114, 2013.

[3] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft, "Privacy-preserving face recognition," in *Proceedings of the International Symposium on Privacy Enhancing Technologies*, 2009.

[4] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg, "Efficient privacy-preserving face recognition," in *Proceedings of the International Conference on Information, Security and Cryptology*, 2010.

[5] Giuseppe Bianchi, Lorenzo Bracciale, and Pierpaolo Loreti, ""better than nothing" privacy with bloom filters: To what extent?," in *Proceedings of the International Conference on Privacy in Statistical Databases*, 2012.

[6] Dan Boneh, Eyal Kushilevitz, Rafail Ostrovsky, and William E. Skeith, "Public key encryption that allows pir queries," in *Proceedings of the International Cryptology Conference, Advances in Cryptology*, 2007.

[7] Martin Beck and Florian Kerschbaum, "Approximate two-party privacy-preserving string matching with linear complexity," in *Proceedings of the IEEE International Congress on Big Data*, 2013, pp. 31–37.

[8] Behrooz Razeghi, Slava Voloshynovskiy, Dimche Kostadinov, and Olga Taran, "Privacy preserving identification using sparse approximation with ambiguization," in *Proceedings of the IEEE International Workshop on Information Forensics and Security*, 2017.

[9] Behrooz Razeghi and Slava Voloshynovskiy, "Privacy-preserving outsourced media search using secure sparse ternary codes," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

[10] Behrooz Razeghi, Slava Voloshynovskiy, Sohrab Ferdowsi, and Dimche Kostadinov, "Privacy-preserving identification via layered sparse code design: Distributed servers and multiple access authorization," in *Proceedings of the European Signal Processing Conference*, 2018, pp. 2578–2582.

[11] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[12] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[13] Florent Perronnin and Christopher Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2007, pp. 1–8.

[14] Ahmet Iscen, Teddy Furon, Vincent Gripon, Michael Rabbat, and Hervé Jégou, "Memory vectors for similarity search in high-dimensional spaces," *IEEE Transactions on Big Data*, 2017.