Agrégation et plongement pour la vérification d'appartenance à un groupe

Marzieh GHEISARI[1], Behrooz RAZEGHI[2], Teddy FURON[1], Laurent AMSALEG[1], Slava VOLOSHYNOVSKIY[2]

[1]Univ Rennes, Inria, CNRS, IRISA
Campus de Beaulieu, Rennes, France

[2]Université of Genève
Carouge, Genève, Suisse

marzieh.gheisari-khorasgani@inria.fr, Behrooz.Razeghi@unige.ch, teddy.furon@inria.fr
laurent.amsaleg@irisa.fr, svolos@unige.ch

Résumé – Ce papier propose un mécanisme pour vérifier l'appartenance à un groupe assurant une certaine sécurité et confidentialité face à un serveur honnête mais curieux. Il s'agit de construire une structure de données enregistrant les signaux des membres tout en empêchant de reconstruire le signal d'un membre en particulier. Ce schéma quantifie les signaux continus en des plongements discrets, et agrège plusieurs plongements ou signaux une seule représentation. Ce sont ces deux mécanismes qui empêchent la reconstruction. Des résultats théoriques et expérimentaux montrent le compromis entre la sécurité et les probabilités d'erreurs de vérification.

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.

1 Introduction

Vérifier qu'un objet / appareil / individu est membre d'un groupe est souvent nécessaire avant de donner l'accès à une ressources sensible. Cependant, l'identification préalable de l'objet / appareil / individu n'est pas nécessaire. On souhaite juste distinguer les membres des non-membres tout en gardant un anonymat relatif. On appèle signal un vecteur extrait d'un objet (PUF passif), d'un appareil (PUF actif), ou d'un individu (données biométriques). Une première phase enrôle les signaux des membres dans une structure de données stockée dans un serveur. Lors de la vérification, le signal d'un client est une requête au système qui autorisera ou non l'accès à la ressource. Pour des raisons de sécurité, la structure doit être protégée. Il faut empêcher un serveur honnête mais curieux de reconstruire les signaux des membres. De même, la vérification ne doit pas révéler l'identité du client.

Le signal client est une version bruitée du signal enrôlé. La vérification doit absorber de telles variations. On suppose que la version bruitée d'un signal est suffisamment dissimilaire et statistiquement indépendante des autres signaux.

Notre approche se compose de deux blocs :

Bloc #1 : Une technique de hashage transforme des vecteurs continus en variables discrètes. Cette perte d'information empêche la reconstruction des signaux.

Bloc #2 : L'agrégation assemble plusieurs signaux en une représentation unique du groupe. Le serveur ne peut rien inférer sur un signal en particulier à partir de cette représentation.

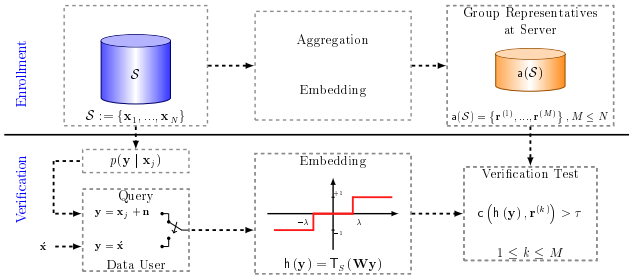

FIGURE 1 – Schéma du modèle proposé.

Ces blocs sont assemblés dans deux configurations : le système hashe les signaux d'abord puis les agrège, ou alors il les agrège d'abord puis hashe ce résultat. A la vérification, le signal requête est toujours hashé avant d'être envoyé au serveur.

2 État de l'art

La vérification d'appartenance à un groupe est généralement faite avec des primitives cryptographiques [1]. Par exemple, un chiffrement homomorphe des signaux permet une comparaison [2] et un seuillage des distances [3, 4] dans le domaine chiffré si le client est actif. On leur reprochera un coût mémoire et complexité important.

L'appartenance à un groupe fait penser au filtre de Bloom en informatique, qui a quelques propriétés de sécurité [5]. Cependant, cette structure de données n'est pas adapté à des signaux continus à moins de les quantifier au préalable [6, 7].

Le plongement de vecteurs est une technique de hashage standard. Le plus adapté à notre scénario est un mécanisme à base de transformée 'parcimonieuse' qui préserve la confidentialité des signaux [8, 9, 10].

L'agrégation de vecteurs est aussi une pratique courante en vision par ordinateur [11, 12, 13]. Cependant, aucune considération de sécurité ou de confidentialité n'est prise en compte dans ces applications. Iscen *et al.* par exemple utilise le paradigme du test par groupe pour assembler plusieurs descripteurs d'image dans un unique vecteur de grande dimension [14]. Ceci est fait de telle manière que la similarité entre un descripteur du groupe et sa représentation est préservée malgré l'aggrégation.

3 Notations et définitions

Les signaux sont des vecteurs dans $\mathbb{R}^d$. Il y a N signaux à enrôler : $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$. Le vecteur $\mathbf{y} \in \mathbb{R}^d$ est le signal requête. La vérification est basée sur deux hypothèses :

$\mathcal{H}_1$: La requête est l'un des signaux du groupe, disons, $\mathbf{y} = \mathbf{x}_j + \mathbf{n}$, avec $\mathbf{n}$ un vecteur bruit.

$\mathcal{H}_0$: La requête est une version bruitée d'un signal n'appartenant pas à $\mathcal{S}$.

À l'enrôlement, l'agrégation s calcule une unique représentation des N signaux : $\mathbf{r} := \mathsf{s}(\mathcal{S})$. La variable ℓ est la taille est bits de cette représentation.

À la vérification, la requête $\mathbf{y}$ est hashée par une fonction h en ℓ bits. Le test prend une décision en comparant $\mathsf{h}(\mathbf{y})$ à $\mathbf{r}$ par la fonction de score c puis seuillage : $t := [\mathsf{c}(\mathsf{h}(\mathbf{y}), \mathbf{r}) > \tau]$.

3.1 Performances en vérification

Les performances sont mesurées par les probabilités de faux négatif $p_{\mathsf{fn}}(\tau) := \mathbb{P}(t = 0|\mathcal{H}_1)$ et faux positif $p_{\mathsf{fp}}(\tau) := \mathbb{P}(t = 1|\mathcal{H}_0)$. Lorsque τ va de $-\infty$ à $+\infty$, ces mesures sont résumées par l'AUC (Area Under Curve). Un autre critère est $p_{\mathsf{fn}}(\tau)$ pour τ t.q. $p_{\mathsf{fp}}(\tau) = \epsilon$, un niveau maximum de faux positif requis dans le cahier des charges.

3.2 Securité et Confidentialité

Un serveur curieux cherche à reconstruire un signal $\mathbf{x}$ à partir de son hash : $\hat{\mathbf{x}} = \mathsf{rec}(\mathsf{h}(\mathbf{x}))$. L'erreur quadratique moyenne atteste de la qualité de la reconstruction : $\mathsf{MSE}_{\mathsf{h}} = \mathbb{E}(\|\mathbf{X} - \mathsf{rec}(\mathsf{h}(\mathbf{X}))\|^2)/d$. La meilleure reconstruction est alors l'espérance conditionnelle : $\hat{\mathbf{x}} = \mathbb{E}(\mathbf{X}|\mathsf{h}(\mathbf{x}))$.

Reconstruire un signal enrôlé à partir de la représentation du groupe est plus aventureux. À cause de l'agrégation, le serveur curieux ne peut reconstruire qu'un seul vecteur $\hat{\mathbf{x}}$ qui sert d'estimation pour tous les signaux du groupe.

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

4 Vérification pour un petit groupe

Cette section étudie les deux configurations pour assembler les blocs #1 et #2 lorsque N est petit.

Bloc #1 : Plongement. Un plongement $\mathsf{h} : \mathbb{R}^d \to \mathcal{A}^\ell$ associe un vecteur à une séquence de ℓ symboles. Nous utilisons celui proposé dans [8, 9]. Il passe $\mathbf{x} \in \mathbb{R}^d$ dans la matrice $\mathbf{W} \in \mathbb{R}^{\ell \times d}$. L'alphabet $\mathcal{A} = \{-1, 0, +1\}$ est imposé en quantifiant les composantes de $\mathbf{W}\mathbf{x}$ d'amplitude plus petite que λ à zéro et les autres à ± 1 suivant leur signe. En espérance, $S = 2d\Phi(-\lambda/\sigma_x)$ symboles sont non nuls.

Bloc #2 : Agrégation. L'agrégation définie $\mathsf{a} : \mathbb{R}^{\ell \times N} \to \mathbb{R}^\ell$ lorsque le bloc #1 est utilisé avant, donnant $\mathsf{s} = \mathsf{a} \circ \mathsf{h}$. Si le bloc #2 est utilisé avant, $\mathsf{s} = \mathsf{h} \circ \mathsf{a}$ et $\mathsf{a} : \mathbb{R}^{d \times N} \to \mathbb{R}^d$.

Dans la première construction, deux stratégies sont

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

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

où $\mathbf{G} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ est une matrice $d \times N$, $\mathbf{G}^\dagger$ est sa pseudo-inverse et $\mathbf{1}_N := (1, \ldots, 1)^\top \in \mathbb{R}^N$. Eq. (2) est appelée 'sum' et (3) 'pinv' dans [14]. Dans la deuxième construction, deux autres stratégies sont la somme (4) et le vote majoritaire (5) :

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

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

Nous avons donc 4 variantes :

— **HoA-2 :** On additionne les signaux (2) et on plonge.
— **HoA-3 :** Ici, l'agrégation (3) précède le plongement.
— **AoH-4 :** On plonge chaque signaux avant l'aggrégation (4).
— **AoH-5 :** Ici, le plongement précède l'agrégation (5).

La fonction score c est toujours $\mathsf{c}(\mathsf{h}(\mathbf{y}), \mathbf{r}) = -\|\mathsf{h}(\mathbf{y}) - \mathbf{r}\|$.

5 Reconstruction et Vérification

On modélise les signaux par $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_d, \sigma_x^2 \mathbf{I}_d)$, et on suppose que $\mathbf{W}$ est une matrice orthogonale carrée. Comme $\mathbf{W}$ préserve la norme, $\mathsf{MSE}_{\mathsf{h}}$ sur $\mathbf{X}$ est la même que l'erreur quadratique moyenne sur $\mathbf{Z} = \mathbf{W}\mathbf{X}$, qui est aussi blanc Gaussien. Par indépendance des composantes de $\mathbf{Z}$, l'espérance conditionnelle est facile à calculer : Soit la densité conditionnée à l'interval $\mathcal{R}_s \subset \mathbb{R}$

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

et les intervalles $\mathcal{R}_0 = [-\lambda, \lambda]$, $\mathcal{R}_1 = (\lambda, +\infty)$ et $\mathcal{R}_{-1} = (-\infty, -\lambda)$. La fonction ϕ_{σ_x} est la densité de $Z \sim \mathcal{N}(0; \sigma_x^2)$ et $\mathbb{1}_{\mathcal{R}_s}$ la fonction indicatrice de $\mathcal{R}_s$.

Sachant que le i-ème symbole de $\mathsf{h}(\mathbf{x})$ égale s signifie que $z_i \in \mathcal{R}_s$. Cette composante est reconstruite en $\hat{z}_i(s) := \mathbb{E}(Z|\mathcal{R}_s)$. On voit que $\hat{z}_i(0) = 0$ par symétrie de $f(z|\mathcal{R}_0)$. Pour $s = 1$, on a $\hat{z}_i(1) = \int_{-\infty}^{+\infty} z.f(z|\mathcal{R}_1)dz = \frac{\sigma_y}{p_1\sqrt{2\pi}} e^{-\frac{\lambda^2}{2\sigma_x^2}}$, où $p_1 :=$

$\mathbb{P}(Z \in \mathcal{R}_1) = \Phi(-\lambda/\sigma_x)$. Par symétrie, $\hat{z}_i(-1) = -\hat{z}_i(1)$, et MSE s'écrit :

$$\mathsf{MSE_h} = \sigma_x^2 \left(1 - \frac{1}{\pi\Phi(-\lambda/\sigma_x)} e^{-\frac{\lambda^2}{\sigma_x^2}}\right) \quad (7)$$

Cette quantité égale $\sigma_x^2(1 - 2\pi^{-1})$ pour $\lambda = 0$. Les plongements sont denses. Toutes les composantes sont reconstruites par $\pm\hat{z}_i$ mais avec une large erreur. Quand λ augmente, cette erreur décroît mais moins de composantes sont reconstruites à une valeur non-nulle. $\mathsf{MSE_h}$ atteint un minimum $\approx 0.19\sigma_x^2$ pour $\lambda \approx 0.60$, où 55% des symboles sont non nuls. Puis, $\mathsf{MSE_h}$ croît jusqu'à σ_x^2 pour λ grand : le plongement devient plus parcimonieux. Quand tout est à zéro, $\mathsf{MSE_h}$ égale σ_x^2.

À partir de l'agrégation $\mathbf{r}$, on reconstruit un unique vecteur $\hat{\mathbf{x}}$, invariant par échelle : multiplier les signaux par un facteur positif ne change pas $\mathbf{r}$. Supposons que l'attaquant reconstruit $\hat{\mathbf{x}} = \kappa\mathbf{u}$. La meilleure remise à l'échelle est $\kappa^\star = \|\mathbf{u}\|^{-2}\mathbf{u}^\top\mathbf{m}$, avec $\mathbf{m} := N^{-1}\sum_{j=1}^N \mathbf{x}_j$. L'attaquant ne connait pas $\kappa^\star$, d'où une erreur de reconstruction plus grande :

$$\mathsf{MSE_s} \geq \sum_{j=1}^N \|\mathbf{x}_j\|^2 - N\frac{(\mathbf{u}^\top\mathbf{m})^2}{\|\mathbf{u}\|^2}. \quad (8)$$

Cette borne est encore minimisée en choisissant $\mathbf{u} \propto \mathbf{m}$. Ainsi, l'agrégation (2) est moins sûre que les autres :

$$\begin{aligned} d.\mathsf{MSE_s} &\geq \mathbb{E}\|\mathbf{X}_j - N^{-1}\mathsf{rec}(\mathsf{a}(\mathcal{S}))\|^2 \quad (9)\\ &= \mathbb{E}\|\mathbf{X}_j - \frac{\mathsf{a}(\mathcal{S})}{N}\|^2 + \frac{\mathbb{E}\|\mathsf{a}(\mathcal{S}) - \mathsf{rec}(\mathsf{a}(\mathcal{S}))\|^2}{N^2}. \end{aligned}$$

Le premier terme correspond à $\|\mathbf{X}_j - \mathbf{m}\|^2$, le second à l'erreur pour inverser le plongement. Au final,

$$\mathsf{MSE_s} \geq \sigma_x^2 \left(1 - \frac{1}{N\pi\Phi(-\lambda/\sigma_x)} e^{-\frac{\lambda^2}{\sigma_x^2}}\right). \quad (10)$$

L'erreur croît avec $N, \forall\lambda \geq 0$: Agréger plus de signaux accroît la sécurité.

5.1 Performance en Vérification

On compare notre schéma à une solution à base de filtre de Bloom réglé de manière optimale ($\ell_B = \lceil N|\log p_{\mathsf{fp}}|\log(2)^{-2}\rceil$). La quantification est la fonction h. Ainsi, un faux négatif se réalise quand $\mathsf{h}(\mathbf{x}_j + \mathbf{n}) \neq \mathsf{h}(\mathbf{x}_j)$.

La figure 2 montre AUC vs. $\mathsf{MSE_h}$ (7) pour différents niveau de parcimonie S/d. Deux schémas se distinguent. À faible sécurité ($\mathsf{MSE_h}$ petit), HoA-3 atteint une plus grande AUC (pour $0.5 \leq S/d \leq 0.6$) ; à forte sécurité, AoH-4 est recommandé (avec $S/d \geq 0.85$). Pour ces régimes, les performances sont meilleures qu'avec un filtre de Bloom.

La figure 3 montre comment les performances décroissent quand le nombre de signatures N augmente. Comme déjà mentionné dans [14], cela dépend du rapport N/d. Plus les signaux sont longs, plus on peut agréger de signaux dans un groupe.

6 Vérification pour plusieurs groupes

Quand N est grand, agréger les signaux dans une représentation unique $\mathbf{r}$ ne fonctionne plus. Nous les répartissons en

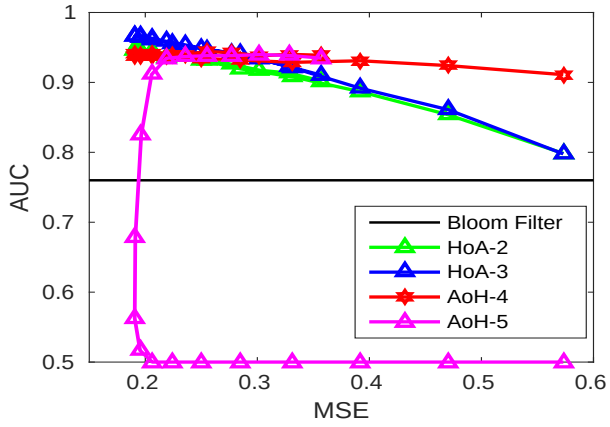

FIGURE 2 – Un seul groupe : AUC vs. $\mathsf{MSE_h}/\sigma_x^2$. $N = 128$, $d = 1024$, $\sigma_n^2 = 0.01$, $S \in (0.1 \times d, 0.9 \times d)$.

$M > 1$ groupes donc M représentations. Ceci est fait aléatoirement ($n = N/M$ signaux par groupe) ou par clustering à l'aide d'un k-means. Sous l'hypothèse $\mathcal{H}_0$, un faux positif se réalise avec probabilité :

$$P_{\mathsf{fp}}(M) = 1 - \prod_{k=1}^M (1 - p_{\mathsf{fp}}^{(k)}). \quad (11)$$

Sous $\mathcal{H}_1$, la requête est liée à un vecteur appartenant à un seul groupe. La probabilité de faux négatif est :

$$P_{\mathsf{fn}}(M) = \sum_{k=1}^M \frac{n_k}{N} p_{\mathsf{fn}}^{(k)} \prod_{l\neq k} (1 - p_{\mathsf{fp}}^{(l)}). \quad (12)$$

Les performances d'un test dépendent principalement de la taille du groupe. Avec la répartition aléatoire, les groupes sont homogènes et partage le même couple $(p_{\mathsf{fp}}, p_{\mathsf{tp}})$.

La figure 4 montre les AUC expérimentale et prédite par (11) et (12) pour M allant de 8 à 512. Comme le clustering fait des groupes non-homogènes en taille, on montre les performances à $n_{min} = \min_{1\leq k\leq M}(n_k)$, où n_k est le taille du k-ème groupe.

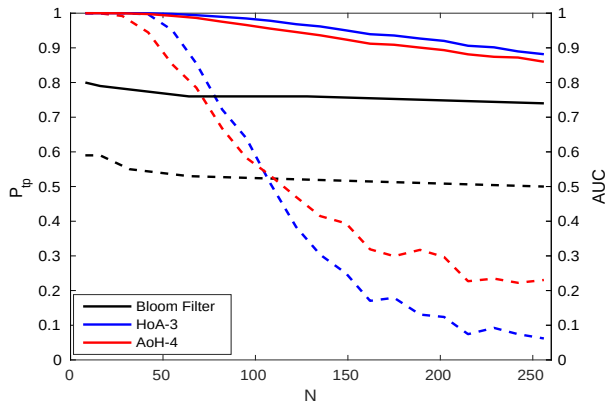

FIGURE 3 – Un seul groupe : AUC and p_{tp} vs. N. AUC (ligne pleine) et $p_{\mathsf{tp}}@p_{\mathsf{fp}} = 10^{-2}$ (ligne en tirets).

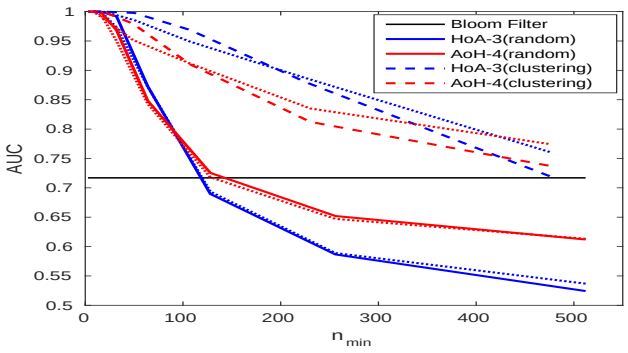

FIGURE 4 – Plusieurs groupes : AUC vs. $n_{\min}$. AUC théorique en pointillé. $N = 4096$, $d = 1024$, $\sigma_n^2 = 10^{-2}$, $S/d = 0.6$ pour HoA-3, et $S/d = 0.85$ pour AoH-4.

Les prédictions sont meilleures avec la répartition aléatoire, mais le clustering améliore les performances en général et plus particulièrement pour HoA-3.

Cependant, la figure 5 montre que cela n'affaiblit pas la sécurité : $\mathsf{MSE_s}$ est légèrement plus petite, mais proche de 1 pour $n_{\min} \geq 100$. Ceci est obtenu pour $M = 32$ pour HoA-3 donnant $\mathsf{AUC} = 0.97$. L'espace est tellement grand que les clusters sont gigantesques et révèlent finalement peu d'information sur les signaux. L'anonymat est réduit car le serveur connait le groupe qui a produit le test positif. Ceci se mesure en terme de k-anonymat par la taille du plus petit groupe , *i.e.* $n_{\min}$.

7 Conclusion

Ce papier propose 4 schémas pour la vérification d'appartenance à un groupe sur des vecteurs réels en grande dimension. L'agrégation et le plongement empêchent la reconstruction précise des signaux enrôlés. L'anonymat est légèrement dévoilé lorsque l'on doit gérer beaucoup de signaux dans plusieurs représentations. Mais l'identité de l'objet / appareil / individu est préservée au sens où il est impossible de reconstruire son signal avec précision.

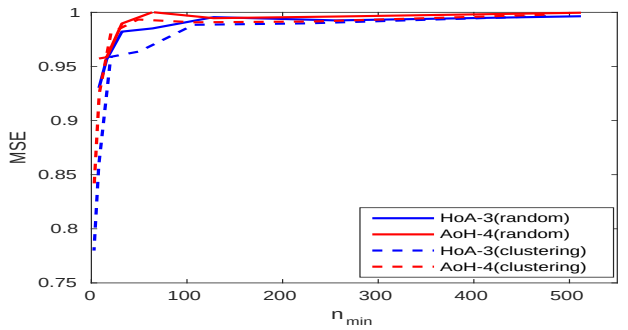

FIGURE 5 – Plusieurs groupes : $\mathsf{MSE_s}$ vs. $n_{\min}$. $N = 4096$, $d = 1024$, $\sigma_n^2 = 10^{-2}$, $S/d = 0.6$ (HoA-3) ou 0.85 (AoH-4).

Références

[1] S. Schechter, T. Parnell, and A. Hartemink, "Anonymous authentication of membership in dynamic groups," in *Int. Conf. on Financial Cryptography*, 1999.

[2] J. R. Troncoso-Pastoriza, D. González-Jiménez, and F. Pérez-González, "Fully private noninteractive face verification," *IEEE Trans. on Information Forensics and Security*, vol. 8, no. 7, pp. 1101–1114, 2013.

[3] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *Int. Symp. on Privacy Enhancing Techno*, 2009.

[4] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *Int. Conf. on Information, Security and Cryptology*, 2010.

[5] G. Bianchi, L. Bracciale, and P. Loreti, ""Better than nothing" privacy with bloom filters : To what extent ?," in *Int. Conf. on Privacy in Statistical Databases*, 2012.

[6] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith, "Public key encryption that allows pir queries," in *Int. Conf. Advances in Cryptology*, 2007.

[7] M. Beck and F. Kerschbaum, "Approximate two-party privacy-preserving string matching with linear complexity," in *Int. Congress on Big Data*, pp. 31–37, 2013.

[8] B. Razeghi, S. Voloshynovskiy, D. Kostadinov, and O. Taran, "Privacy preserving identification using sparse approximation with ambiguization," in *IEEE Int. Work. on Information Forensics and Security*, 2017.

[9] B. Razeghi and S. Voloshynovskiy, "Privacy-preserving outsourced media search using secure sparse ternary codes," in *IEEE ICASSP*, 2018.

[10] B. Razeghi, S. Voloshynovskiy, S. Ferdowsi, and D. Kostadinov, "Privacy-preserving identification via layered sparse code design : Distributed servers and multiple access authorization," in *European Signal Processing Conf.*, pp. 2578–2582, 2018.

[11] J. Sivic and A. Zisserman, "Video google : A text retrieval approach to object matching in videos," in *IEEE Int. Conf. on Computer Vision*, pp. 1470–1477, 2003.

[12] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[13] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *IEEE conf. on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[14] A. Iscen, T. Furon, V. Gripon, M. Rabbat, and H. Jégou, "Memory vectors for similarity search in high-dimensional spaces," *IEEE Trans. on Big Data*, 2017.