HERS: Homomorphically Encrypted Representation Search

Joshua J. Engelsma, Anil K. Jain, and Vishnu Naresh Boddeti

Michigan State University, East Lansing MI 48824, USA
{engelsm7,jain,vishnu}@msu.edu

Abstract. We present a method to search for a probe (or query) image representation against a large gallery in the encrypted domain. We require that the probe and gallery images be represented in terms of a fixed length representation, which is typical for representations obtained from learned networks. Our encryption scheme is agnostic to how the fixed length representation is obtained and can therefore be applied to any fixed length representation in any application domain. Our method, dubbed *HERS* (Homomorphically Encrypted Representation Search), operates by: (i) compressing the representation towards its estimated intrinsic dimensionality, (ii) encrypting the compressed representation using the proposed fully homomorphic encryption scheme, and (iii) searching against a gallery of encrypted representations *directly in the encrypted domain, without decrypting them, and with minimal loss of accuracy.* Numerical results on large galleries of face, fingerprint, and object datasets such as ImageNet show that, for the first time, accurate and fast image search within the encrypted domain is feasible at scale (296 seconds; 46× speed up over state-of-the-art for face search against a background of 1 million).

Keywords: Fixed-Length Representation, Dimensionality Reduction, Intrinsic Dimensionality, Homomorphic Encryption, Privacy-Preserving Search

1 Introduction

In 2014, a hack on the US Office of Personnel Management (OPM) left 22 million user records exposed, including millions of fingerprint records. Data breaches, like the OPM hack, could have untold consequences against those whose personal identifiable information (PII [1]) was compromised, including identity theft, robbery, unauthorized access to secure facilities, and blackmail. Sadly, in today's day and age, data-breaches like these are not isolated incidents [2] [3] [4], motivating and necessitating the development of strong encryption techniques which protect the underlying data *at all times.*

Perhaps the most vulnerable category of stored data still needing adequate protection is that of image representations (*e.g.* face representations). While many forms of data can be sufficiently secured in a database with a well-known

---

[1] https://bit.ly/2HD83Pq   [2] https://wapo.st/39PQuaT   [3] https://wapo.st/2V3kHPS   [4] https://bit.ly/2OQhlM3
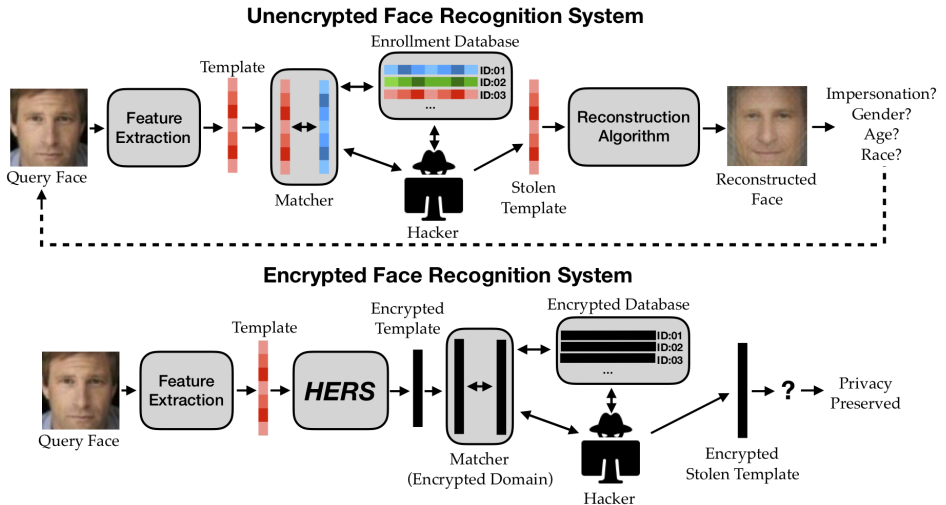
Fig. 1: Overview: Unencrypted and Encrypted face image search systems. Given an image representation (template), it is either (i) stored in a database (enrollment) or (ii) passed to the matcher for searching. Both the database and the matcher are potential points of attack. An attacker can either steal a template directly from an unencrypted database, launching an impersonation attack. Or, if the database were AES encrypted, the hacker could attack the matcher where templates are decrypted for comparison. In our approach, (i) the templates are encrypted during database enrollment and (ii) the templates are matched *within the encrypted domain*. As such, a hacker is unable to exploit any meaningful information from stolen templates.

encryption scheme like the Advanced Encryption Standard (AES) [18], image representations present a unique challenge. In particular, query image representations are often searched against other representations already enrolled in the database (*e.g.* a face search system). Even if the enrolled representations were protected with AES, they would have to be decrypted prior to matching with a query representation, leaving both of them vulnerable at the point of comparison (Fig. 1). As such, it is critically important to develop strong encryption techniques to protect image representations in the database and during matching.

While much research continues in the area of improving the discriminative power of image representations (*e.g.* face recognition [56,42] and image classification [40]), relatively little effort has been invested into ensuring the security of the representations after they have been learned. This is alarming considering deep face representations can be (i) reconstructed back into their corresponding face image (Fig. 1) [29] or (ii) violate user privacy by mining demographic attributes such as age, ethnicity, or gender [27]. More generally, it is well known that local features, such as deep CNN representations [10], SIFT [55,36], HOG [53] and Bag-of-Visual-Words [21], can be inverted back into the image space with high fidelity.

1.1 Homomorphic Encryption

A special class of encryption algorithms which enable basic arithmetic opera-
tions (multiplications and/or additions) in the encrypted domain are known
as homomorphic encryption (HE) systems [16]. Since representations extracted
by CNNs are typically compared using simple distance metrics like euclidean
distance or cosine similarity, HE systems are a plausible solution to protect
the representations within the database, and also during matching. It is worth
noting that HE systems also satisfy other major requirements of feature matching
systems [30], including **diversity**, **revocability**, **security**, **performance**, and
privacy.

The barrier against using HE schemes "off-the-shelf" [13] for protection of
image representations is the computational complexity of arithmetic operations
directly on the ciphertext (in the encrypted domain). This is especially true of
fully-homomorphic encryption schemes (FHE) which enable both additions *and*
multiplications within the encrypted domain [13]. For example, the authors in [49]
showed that a naive implementation of FHE requires 48.7 MB of memory and 12.8
seconds to match a single pair of 512-dimensional encrypted face representations.
Such computational requirements restrict the application of FHE schemes in $1:1$
matching applications and completely render them impractical in $1:m$ search
applications at scale.

To overcome these limitations, we present a practical solution, dubbed HERS,
for $1:m$ encrypted feature matching at scale. This is achieved through a
synergistic combination of dimensionality reduction of features and development
of a more efficient FHE scheme.

1.2 Contributions

- A data encoding scheme that is tailored for efficient $1:m$ representation
  matching in the encrypted domain by leveraging SIMD[5] capabilities of existing
  FHE schemes. Over a gallery of 1 Million 512-dim templates, this provides
  an 11× speed-up.
- A dimensionality reduction scheme, dubbed DeepMDS++, based on Deep-
  MDS [17], a state-of-the-art dimensionality reduction technique. For a 1,536-
  dim ImageNet representation from Inception-Resnet, this provides a 24×
  speed-up at a 1% loss in Average Precision for image retrieval.
- Extensive experimental analysis (face, fingerprint, and ImageNet datasets) in
  terms of accuracy, latency and memory requirements when performing image
  search in the encrypted domain. Our results indicate that *HERS* is the first
  scheme capable of delivering accurate (within ≈ 2% of unencrypted accuracy)
  and real-time (within 5 minutes) image search in the encrypted domain at
  scale (1 million gallery). The overall scheme provides a 46× speed-up over a
  state-of-the-art $1:1$ matching of 128-dim encrypted feature vectors.

---

[5] Single Instruction Multiple Data

2   Related Work

Privacy-Preserving Representations: Methods to secure representations of personally identifying information have been developed over the past decade. These are summarized in [5] into two categories: (i) cryptographic protection, and (ii) pattern recognition based protection. Visual cryptography [32] is a common cryptographic approach for securing biometric data such as fingerprint [45] and face [39] images. Under some schemes, the visual perturbations need to be removed prior to performing a match, exposing the biometric data during authentication or search. Fuzzy-vaults [19] are another cryptosystem which have been utilized for fingerprint [50] and iris [26] recognition. Pattern recognition based protection schemes have been proposed as an alternative to cryptosystems. Examples include non-invertible transformation functions [37] and cancelable biometrics [35]. Additionally, key-binding systems have been proposed to merge a biometric template with a secret key [6,31]. All of these approaches trade-off matching performance for security of the template. In contrast, *HERS* does not suffer from this trade-off i.e., it provides high levels of representation security with minimal loss in matching accuracy.

Encrypted Distance Computation: Homomorphic encryption (HE) cryptosystems enable arithmetic directly on ciphertext, and as such can be leveraged to compute distances between feature vectors entirely in the encrypted domain. However, given the extreme computational complexity of HE [16] most existing works utilizing HE are limited to binary templates and partial homomorphic encryption (PHE) (supports either encrypted additions or encrypted multiplications) [4,25]. A few recent works have have demonstrated the use of Fully Homomorphic Encryption (FHE) schemes [13] (supports encrypted multiplications and additions). Cheon et al. [7] and Boddeti [5] proposed FHE based schemes which leverage a batching technique to reduce the memory and computational requirements for $1:1$ matching of binary iris (Hamming distance) and quantized face representations (cosine distance), respectively, in the encrypted domain. Boddeti showed that two 512-dimensional face representations could be compared with 16 KB of memory and in 2.5 milliseconds. Engelsma et al. [11] adopted the same scheme for $1:1$ matching of 192-dimensional encrypted fingerprint representations in 1.25 milliseconds. In contrast, early FHE schemes for the same took 100 seconds per match [5].

Although these algorithmic savings enable using FHE for $1:1$ matching applications, the time and space complexity remains intractable for most $1:m$ matching applications (image search). For example, encrypted search against a $m = 1,000,000$ gallery representations that are of dimensionality 512 would still take over 230 minutes and 90GB memory with the improved FHE scheme proposed in [5]. As such, in this work we develop a FHE based solution that is explicitly designed for efficient $1:m$ encrypted matching and can be applied to any image search application.

Privacy-Preserving Visual Recognition: Within the broader context of computer vision there is growing interest in privacy-preserving techniques. These
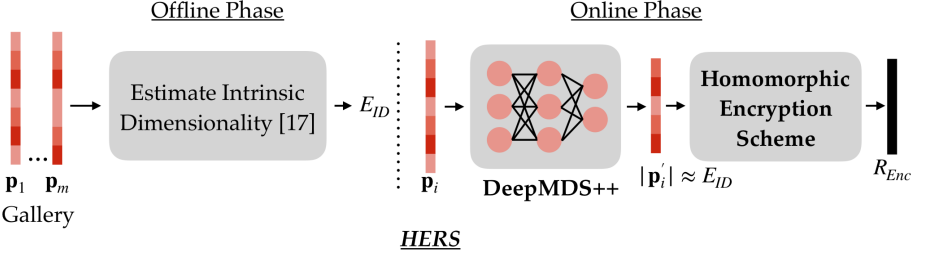
Fig. 2: Schematic Diagram of *HERS*. First, we estimate the intrinsic dimensionality of a given representation [17] in an offline stage from a gallery $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m$. Subsequently, we reduce the dimensionality of the representation towards its intrinsic dimensionality (E_{ID}) as much as possible, such that minimal accuracy is lost, using a deep network based non-linear mapping (DeepMDS++). Finally, the compressed representation $\boldsymbol{p}_i$ is homomorphically encrypted R_{Enc} and passed on to our fast encrypted search algorithm.

approaches are based on cryptographic methods or computer vision techniques. Cryptographic methods include face detection [2,3] using secure multi-party computation, image retrieval [44] by oblivious transfer (a building block of secure multi-party computation), face verification [52] using the Paillier Cryptosystem, video surveillance [51] using Secret Sharing, learning from private data via differential privacy [1] and homomorphic encryption [14,57]. Computer vision techniques include camera localization [46,47] by lifting 2D and 3D points to 2D and 3D lines, detecting private computer screens [24] using CNNs, and activity recognition [38,41] through image manipulation. In contrast to the foregoing, *HERS* adopts a synergistic combination of computer vision techniques in the form of dimensionality reduction and cryptographic methods (FHE scheme), resulting in both efficient and accurate representation matching in the encrypted domain at scale.

3   Approach

There are essentially two ways to improve search efficiency in the encrypted domain: (i) the **encryption scheme** itself can be optimized for faster search, and (ii) the **dimensionality of the representation** can be compressed as far as possible, such that no accuracy is lost. We achieve fast and accurate search at scale in the encrypted domain by coupling both of these techniques in *HERS* (Fig. 2). In the following sections, we elaborate on each of these steps individually.

3.1   Problem Setup

A typical representation matching task involves a database of m template feature vectors $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m] \in \mathbb{R}^{d \times m}$ against which a query representation $\boldsymbol{q} \in \mathbb{R}^d$

is matched. The result of the matching process is a score that determines the degree of similarity between q and each template in P. A common metric that is adopted in computer vision is the Euclidean distance or the cosine similarity. At the core, both of these metrics involve matrix-vector products of the form: $r(P, q) = P^T q$. Hence, the representation matching process is comprised of md scalar multiplications and md scalar additions.

We devise a solution to cryptographically guarantee the security of the database P as well as the query q to prevent leakage of any private information. This can be achieved through a parameterized function that transforms a representation z from the original space into an alternate space, i.e., $\mathcal{E}(z) = f(z; \theta_{pk})$, where $f(\cdot; \theta_{pk})$ is the encryption function with public-key θ_{pk} and $z = g(\mathcal{E}(z); \theta_{sk})$, where $g(\cdot; \theta_{sk})$ is the decryption function with private-key θ_{sk}. The key idea of our paper is to adopt an encryption function to secure the database and the query while retaining our ability to compute their matching score efficiently at scale and without any loss of accuracy i.e.,

$$r(P, q) \approx g\left(r(f(P, \theta_{pk}), f(q, \theta_{pk})); \theta_{sk}\right) \tag{1}$$

FHE satisfies this property and enables us to preserve user privacy. Even if a malicious attacker can gain access to the database of feature representations, without access to the private-key θ_{sk} the attacker cannot reconstruct the underlying image or extract any other information inherent to the representation.

3.2   Fast Secure Distance Computation

We use the Fan-Vercauteren (FV) scheme [12] as our base FHE scheme. The mathematical basis of this scheme lies in modular arithmetic. Building upon this scheme, we propose a data encoding technique that is tailored for efficient $1 : m$ representation matching. We now briefly describe the different components of our approach.

Fan-Vercauteren Scheme: Plaintext space of the FV scheme is represented as a polynomial ring over a finite field $R_t = \mathbb{Z}_t[x]/\Phi_n(x)$, where $t \in \mathbb{Z}$ is an integer and $\Phi_n(x)$ is an irreducible polynomial of degree at most $n - 1$. Upon encrypting the plaintext polynomial, the encrypted numbers (ciphertext) are encoded as polynomials in the ring R_q. The FV scheme utilizes three keys, (1) a private decryption key θ_{sk}, (2) a public encryption key θ_{pk}, and (3) evaluation keys θ_{ev} which are necessary for multiplication over encrypted data. Addition and multiplication of two ciphertexts translates to polynomial addition and dyadic multiplication in R_q. As long as the coefficients of the resulting polynomials do not exceed q, correctness is ensured. The exact description of the entire scheme, including encryption, decryption, ciphertext addition and ciphertext multiplication, is deferred to Appendix A.

Encoding Scheme: The FV scheme is designed to act on integers only. As such, we need to encode our real valued representation $q \in \mathbb{R}^d$ into an integer valued representation $q \in \mathbb{Z}^d$. We quantize a given representation's real-valued features

into integers with a precision of 0.004 and represent these integers in base w. The utility of the FV scheme is critically dependent on the encoding scheme chosen to represent the quantized features in the ring R_t. Therefore, to maintain the integrity of ciphertext computations, the choice of the ring R_t needs to ensure that the range of values after the desired ciphertext operations remain within the same ring.

Our key contribution in this paper is a custom encoding scheme for efficient $1 : m$ matching by utilizing the SIMD primitives [20] of the FV scheme. The primitives operate over an array of numbers instead of a single number, encoding multiple numbers within the same polynomial using the Chinese Remainder Theorem. The encoding scheme in Boddeti [5] is also based on the same principles but is specifically suitable for $1 : 1$ matching.

Given a query $q \in \mathbb{Z}^d$ and a database of feature vectors $P \in \mathbb{Z}^{d \times m}$, the encoding scheme in [5] encodes each sample (column) into a polynomial. In contrast, in *HERS*, the client node encodes each dimension (row) of the representation into a polynomial. A query q is encoded into d plaintexts as,

$$g_i = \sum_{j=1}^{m} q[i] x^{j-1} \ \forall i \in \{1, \ldots, d\} \tag{2}$$

Similarly, for the database, each dimension of all the templates $P_i. \in \mathbb{Z}^m$ are encoded into polynomial, resulting in d plaintexts,

$$h_i = \sum_{j=1}^{m} p_j[i] x^{j-1} \ \forall i \in \{1, \ldots, d\} \tag{3}$$

In this case, the polynomial which encodes the inner products of the query q and the templates P can be obtained as,

$$s = \sum_{i=1}^{d} g_i * h_i = \sum_{j=1}^{n} <q, p_j> x^{j-1} \tag{4}$$

where the product $g * h$ is standard polynomial multiplication. As a consequence, the $m < n^6$ inner products can be computed through d polynomial multiplications. The FV encryption scheme, the corresponding cryptographic primitives, and ciphertext addition and multiplication, operate on this plaintext representation.

Enrollment Protocol: The client generates a public-private key pair. Given a template p, the client (i) quantizes the features, (ii) encodes the quantized template into a plaintext polynomial, (iii) encrypts the plaintext into a ciphertext using the public key, and (iv) transmits the ciphertext along with metadata of the template to the server. The server then adds the encrypted query to the database. Therefore, the server does not have access to the raw representation of the database at any point. The complete enrollment protocol is described in Algorithm 6 of the Appendix.

---

[6] When $m > n$ we can chunk the m samples into $\lceil \frac{m}{n} \rceil$ databases of n samples each.

Table 1: Computational Complexity (# of homomorphic operations) of matching a d dimensional encrypted representation against an encrypted gallery of size m.

Encoding Scheme	Multiplication	Ratio	Addition	Ratio	Rotation	Memory
Naïve	md	1	$m(d-1)$	1	0	$\mathcal{O}(mdn)$
Boddeti [5]	m	d	$m\log_2 d$	$\frac{d-1}{\log_2 d}$	$m\log_2 d$	$\mathcal{O}(mn)$
$HERS$(ours)	$\lceil\frac{m}{n}\rceil d$	$\frac{m}{\lceil\frac{m}{n}\rceil}$	$\lceil\frac{m}{n}\rceil(d-1)$	$\frac{m}{\lceil\frac{m}{n}\rceil}$	0	$\mathcal{O}\left(dn\lceil\frac{m}{n}\rceil\right)$

Search Protocol: Given a query $\boldsymbol{q}$, the client (i) quantizes the features, (ii) encodes the quantized query into a plaintext polynomial, (iii) encrypts the plaintext into a ciphertext using the private key, and (iv) transmits the ciphertext to the server. The server computes the encrypted scores as described above and sends them back to the client. The client now decrypts the encrypted scores using the private-key and obtains the index of the nearest match. This index can then be transmitted back to the server depending on the downstream tasks. Therefore, the server does not have access to the raw representation of the query or the matching scores at any point. The complete search protocol is described in Algorithm 7 of the Appendix.

Computational Complexity The key technical barrier to realizing homomorphic encryption based representation matching is the computational complexity of the FV scheme, especially ciphertext multiplication. Fundamentally, the addition/multiplication of two integers in the plaintext transforms to addition/multiplication of two polynomials of degree n, leading to a n-fold and $\mathcal{O}(n^2)^7$-fold increase in computational complexity for addition and multiplication, respectively. Therefore, mitigating the number of ciphertext multiplications can lead to large gains in practical utility.

Table 1 compares the computational complexity of different encoding schemes for secure distance computation. A naïve implementation of the FV scheme, i.e., no SIMD, encrypts each element of the representation and performs score matching. Such a scheme has a large computational burden bordering on being impractical for real-world applications. The 1:1 matching scheme in [5] is specifically designed for vector-vector inner products by encoding an entire d-dimensional feature vector into a polynomial. A major computational bottleneck in their scheme is the need for expensive ciphertext *rotations* in order to compute the inner product without access to the individual dimensions of the ciphertext vector. Therefore, this approach scales linearly with the size of the database m. In contrast, we observe that our proposed encoding scheme enables $HERS$ to scale to larger databases due to slower rate of increase in computational complexity by a factor of $\sim \mathcal{O}\left(\frac{d}{n}\right)$.

---

[7] In practice, can be reduced to $\mathcal{O}(n\log n)$ through number theoretic transforms.

3.3 Dimensionality Reduction for Faster FHE Matching

From the previous subsection we observe that a smaller feature dimension d provides a greater computational savings for $1:m$ matching in comparison to existing approaches. Therefore, to further ease the computational burden of HERS, we reduce the dimensionality of a given representation as much as possible, while still retaining accuracy (*i.e.* we attempt to map a representation from the ambient space to its intrinsic dimensionality[8]). For example, we show a reduction of an ImageNet representation from its 1,536-dim ambient space to a 64-dim space while losing only 1% average precision, but getting a **24 times** search speed up within the encrypted domain.

Recent work [17] has shown that representations learned by deep convolutional neural networks are highly redundant, i.e., they lie on a low-dimensional manifold whose intrinsic dimensionality is $20\times$ to $30\times$ smaller than the ambient space that the representation is embedded in. The authors in [17] further showed that deep network based non-linear mappings provided a more accurate mapping from a representation's ambient space to its intrinsic space than existing dimensionality reduction techniques. The authors aptly named their dimensionality reduction technique "DeepMDS", since DeepMDS follows a similar paradigm as Multidimensional Scaling (MDS) in preserving pairwise distances between the ambient and intrinsic spaces.

At its core, DeepMDS [17] is a deep network comprised of multiple non-linear layers, and is trained to project an embedding from its ambient space to the intrinsic space, such that pairwise distances are preserved. More formally, let $\mathbf{q} \in \mathbb{R}^d$ be a high-dimensional representation in the ambient space and $f(\cdot; \boldsymbol{w})$ be the DeepMDS non-linear mapping function with parameters $\boldsymbol{w}$. Then, a representation $\mathbf{y}$ in the estimated intrinsic space is computed in accordance with $\mathbf{y} = f(\mathbf{x}; \boldsymbol{w})$

To train the weights $\boldsymbol{w}$ in $f(\cdot; \boldsymbol{w})$, let $\mathbf{G}_1, \mathbf{G}_2, \mathbf{I}_1,$ and $\mathbf{I}_2$ be matrices where rows in each matrix are unit length representations in the ambient space, corresponding rows in $\mathbf{G}_1$ and $\mathbf{G}_2$ are two representations extracted from images with the same class label (genuine pairs), corresponding rows in $\mathbf{I}_1,$ and $\mathbf{I}_2$ are two representations extracted from images with different class labels (imposter pairs), and N is the number of rows in the matrices (mini-batch size). Then, each of the rows of these matrices are projected to the intrinsic space giving four new matrices defined as $\hat{\mathbf{G}}_1, \hat{\mathbf{G}}_2, \hat{\mathbf{I}}_1,$ and $\hat{\mathbf{I}}_2$ with a much lower dimensionality (reduced number of columns).

Next, pairwise distances between genuine pairs D_G and imposter pairs D_I in the ambient space and intrinsic space $(\hat{D}_G, \hat{D}_I)$ are computed as:

$$D_G = u(\mathbf{G_1G_2}^T) \quad D_I = u(\mathbf{I_1I_2}^T) \quad \hat{D}_G = u(\hat{\mathbf{G}}_1\hat{\mathbf{G}}_2^T) \quad \hat{D}_I = u(\hat{\mathbf{I}}_1\hat{\mathbf{I}}_2^T) \quad (5)$$

where $u(\cdot)$ returns the diagonal elements of the matrix multiplication.

Finally, the DeepMDS loss is computed in accordance with:

$$\mathcal{L}_D = |D_G - \hat{D}_G|_2^2 + |D_I - \hat{D}_I|_2^2 \quad (6)$$

---

[8] Intrinsic dimensionality [17] is the lowest number of dimensions needed to maintain the information present in the ambient representation.

Although the authors in [17] showed impressive results in their dimensionality reduction scheme, it suffers from several limitations which needed to be addressed in order to use in *HERS*. First, DeepMDS was mostly trained and tested on the same dataset, making it vulnerable to overfitting. This is problematic as we would almost certainly need to train and test on different datasets in real world applications. Second, the DeepMDS loss is formulated to optimize over the average pairwise distance deviations in a mini-batch. However, this does not focus the training towards smaller subsets or modes of representations for which the embedding is not effective, since they will be averaged out in a randomly selected mini-batch. Both of these weaknesses manifest themselves downstream in the form of lower accuracy at lower dimensions, especially as we noticed, when training and testing on different datasets. Since our application is highly cognizant of accuracy (we do not want to trade security for accuracy), we propose **DeepMDS++**, a deep network based dimensionality reduction technique which aims to solve the short-comings of DeepMDS.

Covariance Penalty: To help prevent overfitting on cross-dataset evaluations, in DeepMDS++, we adopt a covariance penalty to encourage features in the intrinsic space which are less correlated. As was shown in [8], decorrelated features generalize better than correlated features. More formally, in DeepMDS++, we compute an additional loss term $\mathcal{L}_c$ in accordance with the equation below:

$$\mathcal{L}_c = |C - diag(C)|_F^2 \tag{7}$$

where C is the covariance matrix computed across our mini-batch $[\hat{\mathbf{G}}_1; \hat{\mathbf{G}}_2; \hat{\mathbf{I}}_1; \hat{\mathbf{I}}_2]$ and $diag(\cdot)$ returns a diagonal matrix, with diagonal values equivalent to the input matrix.

Hard Pair Mining: To prevent DeepMDS from overfitting to a mode of representations, via its averaging of the pairwise distance deviations in a mini-batch, we introduce a hard-mining strategy [42] with a specific focus on "harder pairs" to fill out our mini-batch. In particular, we want to select those pairs for which the pairwise distance is not well preserved after their mapping into the intrinsic space. In this way we prevent DeepMDS++ from doing well on the average case, while ignoring other modes in the data (which seriously effects cross dataset generalization).

More formally, the mini-batch indices of the hard genuine pairs P_G and the hard imposter pairs P_I can be computed as:

$$P_G = argsort(D_G - \hat{D}_G) \quad P_I = argsort(D_I - \hat{D}_I) \tag{8}$$

where $argsort(\cdot)$ will return the indices of the rows in matrices $\mathbf{G}_i$ and $\mathbf{I}_i$, $i \in \{1, 2\}$ with the highest deviation in pairwise distances following their projection into the intrinsic space (i.e. the hardest pairs).

In our experimental results (all cross-dataset) we show that the incorporation of these training techniques into DeepMDS++ aids us in compressing representations towards their intrinsic dimensionality with lower loss of accuracy, ultimately enabling us to perform matching in the encrypted domain at much faster speeds.

4 Experiments

Our experiments are designed to (i) evaluate the improvement in image search speed with our improved fully homomorphic encryption wrapper, (ii) demonstrate the further speed improvements following dimensionality reduction, and (iii) show the improvements of our deep dimensionality reduction method **DeepMDS++** over DeepMDS, enabling our compression of representations for faster encrypted search.

4.1 Implementation Details

DeepMDS++ is implemented in Tensorflow 1.14.0. The network architecture is provided in Appendix D. We use the Adam optimizer [23] with a starting learning rate of 3×10^{-4} and weight regularization with decay set to 4×10^{-5}. We train for 250 epochs, lasting 6 hours, on a NVIDIA GeForce RTX 2080 Ti GPU. When using hard-mining, we set the mini-batch size $N = 4,000$. At the start of training, we mine 50 hard genuine pairs and 50 hard imposter pairs which are augmented with 200 random genuine pairs and 200 random imposter pairs, respectively. Then, we linearly increase the number of hard genuine and hard imposter pairs from 50 to 250 over the 250 training epochs.

The implementation of the homomorphic encryption component of *HERS* is based on the SEAL library [43]. For all experiments, we used a 10-core Intel i9-7900X processor running at 3.30 GHz. We use a single-threaded environment. The three main parameters of the encryption scheme (n, t, q) are set to $n = 4,096$, $t = 1,032,193$ and q is the default value[9] in SEAL i.e., a product of 3 very large primes, each 36 bits long.

4.2 Evaluation Datasets

- **ImageNet ILSVRC 2012:** 1000 classes with 1.28 million training images and 50K validation images. We randomly select 100 classes from the training and validation set for training/testing classification accuracy and precision/recall, and we use the entire validation set for testing precision @ 10.
- **FaceScrub + MegaFace [22]:** MegaFace: 1 million distractor faces; FaceScrub: $\approx 3.5K$ celebrity faces. These are commonly coupled datasets for evaluating face search performance at scale.
- **CASIA [56]:** 450K face images from 10K subjects. 100K subset used to train DeepMDS++ prior to its application on MegaFace and FaceScrub.
- **NIST SD4 [54] + MSP 1.1 Million [58]:** NIST SD4: Contains 2,000 probe/mate inked-rolled fingerprint pairs. MSP: privately held forensic database with over 1.5 million rolled fingerprints (1.1 million used as distractors; a separate 100K used to train DeepMDS++).

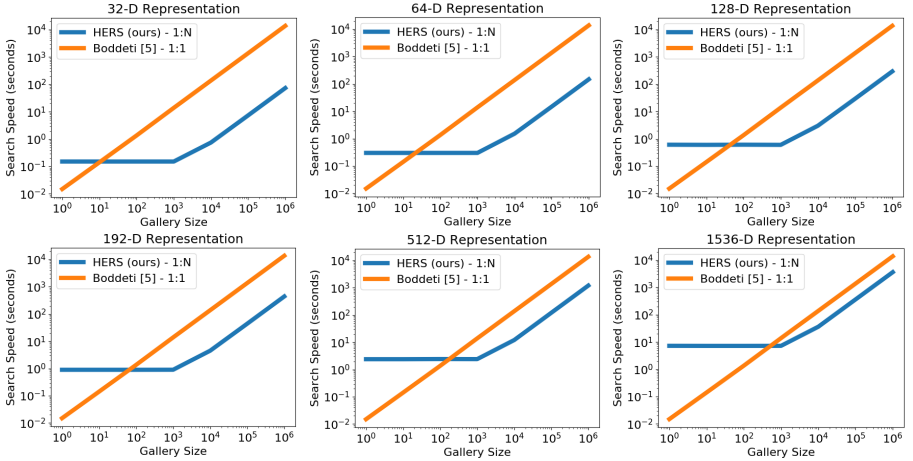[9] In practice, much smaller values of q are sufficient for our purpose.

Fig. 3: Computational complexity comparison (log-log scale) of *HERS* with Boddeti [5] for $1 : m$ matching as a function of gallery size m and different representation dimensionality.

4.3   Representation Models

- **Inception ResNet v2 [48]:** Combines inception modules with residual connections to achieve one of the highest performing results on ImageNet 2012 (Top-1 accuracy 80.3%). We use a pre-trained model[10] to extract ImageNet training and validation features (1,536-dim representations).
- **ArcFace [9]:** Obtains state-of-the-art results on the MegaFace Challenge via (i) architectural refinements, (ii) cleaned training data, and (iii) an additive arc margin loss. We extracted 512-dim embeddings using a publicly available [11], pre-trained ArcFace model.
- **DeepPrint [11]:** One of only several DCNNs for extracting deep representations from fingerprints (192-dim features for DeepPrint). DeepPrint matches the accuracy of state-of-the-art commercial matchers via integration of fingerprint domain knowledge during training.

4.4   Evaluation Protocol and Experimental Results

To evaluate the efficacy of *HERS*, (i) we benchmark its efficiency at different gallery sizes and with different representation dimensions against Boddeti [5] , and (ii) we benchmark the matching and search accuracy at different representation dimensions against DeepMDS [17].

Efficiency: Figure 3, compares the search speed of the encryption component of *HERS* with [5]. We show the speed for 1,536-dim, 512-dim, and 192-dim to align with the original dimensionality of our different representation models.

---

[10] https://keras.io/applications/#inceptionresnetv2
[11] https://github.com/deepinsight/insightface#pretrained-models

Table 2: Computational Complexity of Search Against 1 Million Gallery

Dimension Method	Time (seconds)						Memory (GB)					
	32	64	128	192	512	1536	32	64	128	192	512	1536
Boddeti [5]			13,825						90			
HERS (Ours)	74	148	296	444	1,212	3,635	1.4	2.8	5.5	8.3	22	68

Table 3: ImageNet

(a) Average Precision[a]

Dimension	DeepMDS	DeepMDS++
128	0.92	0.92
64	0.89	**0.91**
32	0.68	**0.88**
16	0.37	**0.78**

[a] Original AP of 0.92 at 1536-dim

(b) Top-1 Accuracy[a](%)

Dimension	DeepMDS	DeepMDS++
128	85.4	**85.7**
64	83.4	**85.0**
32	66.1	**81.8**
16	42.7	**71.9**

[a] Original accuracy of 86.2% at 1536-dim

Subsequently, we show the speed up when compressing those representations to dimensions of 128, 64, or 32. This highlights the importance of combining encryption techniques together with dimensionality reduction in order to perform encrypted search at scale. The results indicate that at small gallery sizes 1:1 matching from Boddeti [5] is unsurprisingly more efficient, since it was explicitly designed for 1:1 matching. However, as the gallery increases *HERS* is more efficient. Admittedly these methods are orders of magnitude slower than matching in the unencrypted domain, but *HERS* is the first practically scalable search over an encrypted database. In terms of memory, for a 64-dim representation, *HERS* requires 5.9 MB for the probe template and 2.8 GB for a gallery of size 1 million. In comparison [5] requires less memory for the probe at 19K, but exhausts 90 GB for the same gallery size. Table 2 reports the computational complexity of encrypted search, in terms of time and memory, against a gallery of 1 Million for different representation dimensionality.

Accuracy: We evaluate the accuracy of DeepMDS++, the dimensionality reduction component of *HERS* in terms of: (i) Top-1 classification accuracy and Precision-Recall curves from a subset of 100 classes of the ImageNet validation set, (ii) Precision @ 10 (Precision in Top-10 retrieved samples) using the entire ImageNet validation set, (iii) Rank-1 Face Search Performance on MegaFace (1 million distractors), and (iv) Rank-1 Fingerprint Search performance (using NIST SD4 against a gallery of 1.1 million [58]). For classification experiments on ImageNet, we train a Linear SVM classifier (one-vs-rest) on top of our embeddings. To compute Precision @ 10, we randomly select 10 probes from each validation class, and use the remaining 40 from each class as mates. These mates are combined with the remaining 45,000 distractors with 900 classes and 50 images / class.
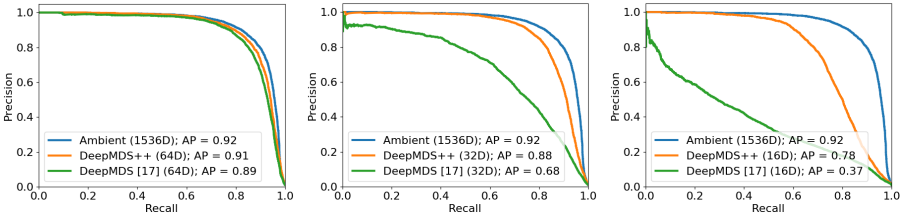
Fig. 4: Precision-Recall curves using Inception ResNet V2 on ImageNet 2012

Table 4: Face and Fingerprint Search: Rank-1 Accuracy (%)

(a) MegaFace (Gallery: 1 Million)   **(b)** Fingerprint (Gallery: 1.1 Million)

512-dim	256-dim	128-dim	64-dim	192-dim	64-dim	32-dim	16-dim
81.4	81.4	78.8	66.2	94.5	94.2	94.0	87.5

Table 3a and Table 3b compares the performance of DeepMDS++ and DeepMDS on the ImageNet dataset for the task of image retrieval and image classification, respectively. Figure 4 shows the full precision recall curves at dimensions 1536, 64, 32 and 16 for Inception ResNet v2 embeddings. The results indicate that,

- It is feasible to compress the image representations by large factors for a small performance penalty. For instance, the representation can be compressed by a factor of $24\times$ (1536-dim to 64-dim) for a performance loss of 1% (92% to 91%) in AP and 1% (86.2% to 85.0%) Top-1 accuracy or by a factor of $48\times$ (1536-dim to 32-dim) for a performance loss of 4% (92% to 88%) in AP and 4.5% (86.2% to 81.8%) in Top-1 accuracy.
- Representations compressed through DeepMDS++ are able to retain more discriminative information at lower dimensions compared to DeepMDS, especially at lower dimensions. For instance, at 32-dim DeepMDS++ obtains an average precision of 88% compared to 68% by DeepMDS for image retrieval. Similarly DeepMDS++ Top-1 image classification accuracy is 81.8% compared to 66.1% for DeepMDS.
- From the precision recall curves, we note that as the Inception ResNet v2 ambient embeddings are compressed to a lower number of dimensions, DeepMDS++ has an increasing advantage over the original DeepMDS [17].

Table 4a and Table 4b show the Rank-1 accuracy of encrypted face (MegaFace [22]) and fingerprint (MSP [58]) search, respectively, as we compress the representations with DeepMDS++. These results suggest that, practically speaking, we can compress the face representations by a factor of $4\times$ (512-dim to 128-dim) for a performance loss of 2.6% (81.4% to 78.8%). Similarly fingerprint representations can be compressed by a factor of $6\times$ (192-dim to 32-dim) for a performance loss of 0.5% (94.5% to 94.0%). The degree of compression of the representation

Table 5: Precision @ Rank 10 on ImageNet ILSVRC-2012 Validation Set (%)

Dimensionality[1]	**DeepMDS++** (proposed)	w/o Hard Mining	w/o $\mathcal{L}_c$ w/o Hard Mining (DeepMDS [17])
256	**67.6**	66.1	66.1
128	**66.5**	64.1	64.3
64	**64.0**	60.1	58.9
32	**57.6**	43.6	36.4
16	**38.0**	17.0	17.1

[1] Performance with original 1536-dim Inception ResNet V2 features is 69.7%.

by DeepMDS++ closely mirrors the intrinsic dimensionality estimates[12] of the respective representations. The intrinsic dimensionalities of ArcFace, DeepPrint and Inception ResNet V2 are 15, 5 and 6 respectively, suggesting that unfolding ArcFace down to 16-dim is about three times as hard as compressing the latter two representations.

Ablation: Table 5 shows the impact of hard-pair mining and the covariance loss on the ability of DeepMDS to compress the representations. We observe that hard-pair mining is effective across all dimensions while the covariance loss is more effective around 32 to 64 dimensions where a noticeable benefit is observed. The ability of DeepMDS++ to retain more discriminative information than DeepMDS affords compression to lower dimensions which in turn synergistically aids in improving the efficiency of encrypted search.

5   Discussion

Here we briefly comment on our choice of cryptographic solution, namely, FHE, the limitations induced by our choice and contrast it with other plausible cryptographic solutions. Two other alternative cryptographic solutions that can be employed in lieu of or in conjunction with FHE are:

Partial Homomorphic Encryption (Paillier Cryptosystem [34]): This scheme supports only homomorphic additions and is significantly more efficient for scalars than the FV scheme we use. However, it does not support massively vectorized SIMD operations, which is the key source of efficiency in *HERS*.

Secure Multi-Party Computation [15]: This scheme can be employed to securely compute the nearest neighbors (including matching score and *argmax* index) by employing multiple parties that communicate secret shares with each other in such a way that no single party can access all the features. This approach trades-off low computation for high communication costs. Furthermore, it requires that the database be split among multiple parties which may not be desirable in some applications. In contrast, the FHE scheme trades-off low-communication

---

[12]   ID estimate code: `https://github.com/human-analysis/intrinsic-dimensionality`

costs for high computational costs. The main drawback of the FV scheme is the limited arithmetic operations supported by it, namely addition and multiplication. Therefore, computing non-linear functions like *max* and *argmax* are not supported by *HERS*. In the context of search, as opposed to 1:1 verification, it is often sufficient to protect the gallery and query representations, as opposed to the matching scores. In such cases, our solution of computing the matching scores in the encrypted domain, having the client decrypt the scores and finally having the server respond with the matched database index should suffice.

6   Conclusions

In this paper, we proposed *HERS*, a scheme for accurate and practical search over homomorphically encrypted representations at scale. The efficiency of *HERS* stems from (i) efficient cryptographic primitives for encrypted matrix-vector products, and (ii) DeepMDS++, a non-linear dimensionality reduction technique to reduce operations in the encrypted domain. The accuracy of *HERS* stems from (i) the exact computations of our cryptographic primitive, and (ii) the effectiveness of DeepMDS++ in maintaining matching performance at large compression factors. Our experimental results demonstrate, for the first time, practical (under 5 minutes) and accurate (within $\approx 2\%$ of unencrypted accuracy) image search (for face, fingerprint and ImageNet) against 1 Million gallery in the encrypted domain.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318 (2016)
2. Avidan, S., Butman, M.: Blind vision. In: European Conference on Computer Vision (2006)
3. Avidan, S., Butman, M.: Efficient methods for privacy preserving face detection. In: Advances in Neural Information Processing Systems (2007)
4. Barni, M., Droandi, G., Lazzeretti, R.: Privacy protection in biometric-based recognition systems: A marriage between cryptography and signal processing. IEEE Signal Processing Magazine **32**(5), 66–76 (2015)
5. Boddeti, V.N.: Secure face matching using fully homomorphic encryption. In: IEEE Conference on Biometrics Theory, Applications and Systems (2018)
6. Boddeti, V.N., Kumar, B.V.: A framework for binding and retrieving class-specific information to and from image patterns using correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(9), 2064–2077 (2012)
7. Cheon, J.H., Chung, H., Kim, M., Lee, K.W.: Ghostshell: Secure biometric authentication using integrity-based homomorphic evaluations. IACR Cryptology ePrint Archive **2016**, 484 (2016)
8. Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., Batra, D.: Reducing overfitting in deep networks by decorrelating representations. arXiv preprint arXiv:1511.06068 (2015)

9. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
10. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
11. Engelsma, J.J., Cao, K., Jain, A.K.: Learning a fixed-length fingerprint representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
12. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive **2012**, 144 (2012)
13. Gentry, C., Halevi, S.: Implementing gentrys fully-homomorphic encryption scheme. In: International Conference on the Theory and Applications of Cryptographic Techniques (2011)
14. Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., Wernsing, J.: CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning (2016)
15. Goldreich, O.: Secure multi-party computation. Manuscript. Preliminary version **78** (1998)
16. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge university press (2009)
17. Gong, S., Boddeti, V.N., Jain, A.K.: On the intrinsic dimensionality of image representations. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
18. Heron, S.: Advanced encryption standard (AES). Network Security **2009**(12), 8–12 (2009)
19. Juels, A., Sudan, M.: A fuzzy vault scheme. Designs, Codes and Cryptography **38**(2), 237–257 (2006)
20. Juvekar, C., Vaikuntanathan, V., Chandrakasan, A.: *GAZELLE*: A low latency framework for secure neural network inference. In: USENIX Security Symposium). pp. 1651–1669 (2018)
21. Kato, H., Harada, T.: Image reconstruction from bag-of-visual-words. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
22. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Korayem, M., Templeman, R., Chen, D., Crandall, D., Kapadia, A.: Enhancing lifelogging privacy by detecting screens. In: CHI Conference on Human Factors in Computing Systems (2016)
25. Lagendijk, R.L., Erkin, Z., Barni, M.: Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation. IEEE Signal Processing Magazine **30**(1), 82–105 (2012)
26. Lee, Y.J., Park, K.R., Lee, S.J., Bae, K., Kim, J.: A new method for generating an invariant iris private key based on the fuzzy vault system. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **38**(5), 1302–1313 (2008)
27. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: IEEE International Conference on Computer Vision (2015)
28. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: International Conference on the Theory and Applications of Cryptographic Techniques (2010)

29. Mai, G., Cao, K., Yuen, P.C., Jain, A.K.: On the reconstruction of face images from deep face templates. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(5), 1188–1202 (2018)
30. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer Science & Business Media (2009)
31. Nandakumar, K., Nagar, A., Jain, A.K.: Hardening fingerprint fuzzy vault using password. In: International Conference on Biometrics (2007)
32. Naor, M., Shamir, A.: Visual cryptography. In: Workshop on the Theory and Application of of Cryptographic Techniques (1994)
33. Oded Goldreich: Foundations of Cryptography: Basic Applications. Cambridge University Press (2004). https://doi.org/10.1561/0400000001
34. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 223–238 (1999)
35. Patel, V.M., Ratha, N.K., Chellappa, R.: Cancelable biometrics: A review. IEEE Signal Processing Magazine **32**(5), 54–65 (2015)
36. Pittaluga, F., Koppal, S.J., Kang, S.B., Sinha, S.N.: Revealing scenes by inverting structure from motion reconstructions. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
37. Ratha, N.K., Connell, J.H., Bolle, R.M.: Enhancing security and privacy in biometrics-based authentication systems. IBM systems Journal **40**(3), 614–634 (2001)
38. Ren, Z., Jae Lee, Y., Ryoo, M.S.: Learning to anonymize faces for privacy preserving action detection. In: European Conference on Computer Vision (2018)
39. Ross, A., Othman, A.: Visual cryptography for biometric privacy. IEEE Transactions on Information Forensics and Security **6**(1), 70–81 (2010)
40. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
41. Ryoo, M.S., Rothrock, B., Fleming, C., Yang, H.J.: Privacy-preserving human activity recognition from extreme low resolution. In: AAAI Conference on Artificial Intelligence (2017)
42. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
43. Microsoft SEAL (release 3.4). `https://github.com/Microsoft/SEAL` (Oct 2019), microsoft Research, Redmond, WA.
44. Shashank, J., Kowshik, P., Srinathan, K., Jawahar, C.: Private content based image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
45. Soutar, C., Roberge, D., Stoianov, A., Gilroy, R., Kumar, B.V.: Biometric encryption. In: ICSA Guide to Cryptography, vol. 22 (1999)
46. Speciale, P., Schonberger, J.L., Kang, S.B., Sinha, S.N., Pollefeys, M.: Privacy preserving image-based localization. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
47. Speciale, P., Schonberger, J.L., Sinha, S.N., Pollefeys, M.: Privacy preserving image queries for camera localization. In: IEEE International Conference on Computer Vision (2019)
48. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI Conference on Artificial Intelligence (2017)

49. Troncoso-Pastoriza, J.R., González-Jiménez, D., Pérez-González, F.: Fully private noninteractive face verification. IEEE Transactions on Information Forensics and Security **8**(7), 1101–1114 (2013)
50. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy vault for fingerprints. In: International Conference on Audio-and Video-Based Biometric Person Authentication. pp. 310–319 (2005)
51. Upmanyu, M., Namboodiri, A.M., Srinathan, K., Jawahar, C.: Efficient privacy preserving video surveillance. In: International Conference on Computer Vision (2009)
52. Upmanyu, M., Namboodiri, A.M., Srinathan, K., Jawahar, C.: Blind authentication: a secure crypto-biometric verification protocol. IEEE Transactions on Information Forensics and Security **5**(2), 255–268 (2010)
53. Vondrick, C., Khosla, A., Malisiewicz, T., Torralba, A.: Hoggles: Visualizing object detection features. In: IEEE International Conference on Computer Vision (2013)
54. Watson, C.I., Wilson, C.: Nist special database 4. Fingerprint Database, National Institute of Standards and Technology **17**(77) (1992)
55. Weinzaepfel, P., Jégou, H., Pérez, P.: Reconstructing an image from its local descriptors. In: IEEE Conference on Computer Vision and Pattern Recognition (2011)
56. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
57. Yonetani, R., Naresh Boddeti, V., Kitani, K.M., Sato, Y.: Privacy-preserving visual learning using doubly permuted homomorphic encryption. In: IEEE International Conference on Computer Vision (2017)
58. Yoon, S., Jain, A.K.: Longitudinal study of fingerprint recognition. Proceedings of the National Academy of Sciences **112**(28), 8555–8560 (2015)

Appendix

In this Appendix, we include, (a) details of the base homomorphic encryption scheme [12] in Section A, (b) detailed algorithms for the enrollment and search phase of *HERS* in Section B, (c) security analysis of *HERS* in Section C, and (d) architectural details of DeepMDS++ in Section D.

A Fully Homomorphic Encryption

For completeness, we describe the Fan-Vercauteren [12] scheme and the associated homomorphic operations, *i.e.*, ciphertext addition and multiplication. These operations will be used in the enrollment and search phase of *HERS*.

Mathematical Notation: For $t \in \mathbb{Z}$ a ring $R_t = \mathbb{Z}_t[x]/(x^n + 1)$ represents polynomials of degree less than n with the coefficients modulo t. The operators $\lfloor \cdot \rfloor$, $\lceil \cdot \rceil$ and $\lfloor \cdot \rceil$ denote rounding down, up and to the nearest integer respectively. The operator $[\cdot]$ denotes the reduction of an integer by modulo t, where the reductions are performed on the symmetric interval $[-t/2, t/2)$. The operators when applied to a polynomial are assumed to act independently on the coefficients of the polynomial. $a \xleftarrow{\$} \mathcal{S}$ denotes that a is sampled uniformly from the finite set $\mathcal{S}$. Similarly, $a \longleftarrow \chi$ denotes that a is sampled from a discrete truncated

Gaussian. We note the plaintext polynomial (also called message) as m and the ciphertext polynomial as ct.

Fan-Vercauteren Scheme [12]: The FV scheme encodes integers to polynomials in a ring R_t (see Eq.2 and Eq.3 for our encoding), referred to as plaintext. Given such a polynomial plaintext, the FV scheme defines the encryption and decryption protocols for such polynomials. The ciphertext is encoded as polynomials in a different ring R_q.

Let λ be the desired level of security, w the base to represent numbers in, and $l = \lfloor \log_w q \rfloor$ the number of terms in the decomposition of q into base w. Below are the details of the FV scheme in terms of key generation, encryption, decryption, addition and multiplication over encrypted integers.

Algorithm 1 Key Generation

1: **procedure** GETKEYS(λ, l, w, q)
2:     Sample: $\boldsymbol{\theta}_{sk} \xleftarrow{\$} R_2$ ▷ private (secret) key
3:     Sample: $\boldsymbol{a} \xleftarrow{\$} R_q$ and $\boldsymbol{e} \longleftarrow \chi$
4:     $\boldsymbol{\theta}_{pk} = ([-(\boldsymbol{a}\boldsymbol{\theta}_{sk} + \boldsymbol{e})]_q, \boldsymbol{a})$ ▷ public key
5:     $\boldsymbol{\theta}_{ev} = \emptyset$
6:     **for** $i = 1$ to l **do**   ▷ generate evaluation keys
7:         Sample: $\boldsymbol{a}_i \xleftarrow{\$} R_q$, $\boldsymbol{e}_i \longleftarrow \chi$
8:         $\boldsymbol{\theta}_{ev}^i = ([-(\boldsymbol{a}_i\boldsymbol{\theta}_{sk} + \boldsymbol{e}_i) + w^i\boldsymbol{\theta}_{sk}^2]_q, \boldsymbol{a}_i)$
9:         $\boldsymbol{\theta}_{ev} = \boldsymbol{\theta}_{ev} \cup \{\boldsymbol{\theta}_{ev}^i\}$
10:     **end for**
11:     return $\boldsymbol{\theta}_{pk}$, $\boldsymbol{\theta}_{sk}$, $\boldsymbol{\theta}_{ev}$ ▷ return all the keys
12: **end procedure**

Algorithm 2 Encryption

1: **procedure** ENCRYPT(m, $\boldsymbol{\theta}_{pk}$, q, t)
2:     Sample: $\boldsymbol{u} \xleftarrow{\$} R_2$, $\boldsymbol{e}_1 \longleftarrow \chi$ and $\boldsymbol{e}_2 \longleftarrow \chi$
3:     $\Delta = \lfloor \frac{q}{t} \rfloor$
4:     $ct = ([\Delta m + \boldsymbol{\theta}_{pk}[0]\boldsymbol{u} + \boldsymbol{e}_1]_q, [\boldsymbol{\theta}_{pk}[1]\boldsymbol{u} + \boldsymbol{e}_2]_q) = (\boldsymbol{ct}[0], \boldsymbol{ct}[1])$
5:     return ct
6: **end procedure**

Algorithm 3 Decryption

1: **procedure** DECRYPT(ct, θ_{sk}, q, t)
2:     $pt = \left[\left\lfloor \frac{t}{q} \left[ct[0] + ct[1]\theta_{sk} \right]_q \right\rceil \right]_t$
3:     return pt
4: **end procedure**

Algorithm 4 Ciphertext Addition

1: **procedure** CIPHERADD(ct_0, ct_1, q)
2:     $ot = ([ct_0[0] + ct_1[0]]_q, [ct_0[1] + ct_1[1]]_q)$
3:     return ot
4: **end procedure**

Algorithm 5 Ciphertext Multiplication

1: **procedure** CIPHERMULTIPLY(ct_0, ct_1, w, l, q, t)
2:     $c_0 = \left[\left\lfloor \frac{t}{q}(ct_0[0]ct_1[0]) \right\rceil \right]_q$
3:     $c_1 = \left[\left\lfloor \frac{t}{q}(ct_0[0]ct_1[1] + ct_0[1]ct_1[0]) \right\rceil \right]_q$
4:     $c_2 = \left[\left\lfloor \frac{t}{q}(ct_0[1]ct_1[1]) \right\rceil \right]_q$
5:     $c_2 = \sum_{i=0}^{l} c_2^{(i)} w^i$ and compute
6:     $c_0' = c_0 + \sum_{i=1}^{l} \theta_{ev}[i][0]c_2^{(i)}$
7:     $c_1' = c_1 + \sum_{i=1}^{l} \theta_{ev}[i][1]c_2^{(i)}$
8:     return (c_0', c_1')
9: **end procedure**

B   Protocols

Here we describe the detailed algorithms of the two phases in *HERS*, namely, *enrollment* (Algorithm 6) and *search* (Algorithm 7). Both of these algorithms are built upon the cryptographic primitives described in Section A.

B.1   Enrollment

Algorithm 6 describes our entire enrollment procedure. The algorithm is designed to handle the scenario where the number of samples in the database m is larger than the ring dimension n (degree of the polynomial). The algorithm also considers a more practical scenario of online enrollment, i.e., we may wish to enroll one

gallery feature vector at a time to the encrypted database. For this purpose, we first encrypt an all-zero feature representation and update it with each new gallery we wish to enroll. This is implemented in Lines 12 - 22 of Algorithm 6 below.

Algorithm 6 *HERS* Enrollment

1: **Encryption Parameters:** coefficient bit length b_c, plaintext modulus t, ciphertext modulus q, ring dimension n
2: Server initializes empty database $\mathcal{D}_i \leftarrow \emptyset \; \forall i \in \{1, \ldots, d\}$, label set $\mathcal{I} \leftarrow \emptyset$ and database index $k = 0$, $v = 0$
3: $\boldsymbol{\theta}_{pk}, \boldsymbol{\theta}_{sk}, \boldsymbol{\theta}_{ev} = GetKeys(\lambda, l, w, q)$ ⊳ client generates keys
4: **Inputs:** $\boldsymbol{id} \in \mathbb{N}^m$ and $\boldsymbol{q} \in \mathbb{R}^{d \times m}$ ⊳ m feature vectors of dimension d each
5: **for** $i = 1$ to d **do**
6:      $v \leftarrow k \bmod n$
7:      $\boldsymbol{m} \leftarrow BatchEncode(\boldsymbol{q}_{i \cdot}; v)$ ⊳ i-th dim to v-th index of plaintext
8:      $\boldsymbol{ct}_i = Encrypt(\boldsymbol{m}; \boldsymbol{\theta}_{pk}, \text{q,t})$
9: **end for**
10: $k \leftarrow k + m$ ⊳ increment database index
11: Send $(\{\boldsymbol{ct}_1, \ldots, \boldsymbol{ct}_d\}, \boldsymbol{id})$ to the server
    ⊳ enrollment at server
12: $\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{id}\}$
13: $\tilde{v} \leftarrow \lceil \frac{k}{n} \rceil$
14: **if** $\tilde{v} > v$ **then**
15:      $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d\}$
16:      $v \leftarrow \tilde{v}$
17:      $\boldsymbol{r}_i \leftarrow Encrypt(\boldsymbol{0}; \boldsymbol{\theta}_{pk}, \text{q,t}) \; \forall i \in \{1, \ldots, d\}$ ⊳ initialize all zero ciphertext
18: **else**
19:      **for** $i = 1$ to d **do** ⊳ enrollment at server
20:          $\boldsymbol{r}_i \leftarrow CipherAdd(\boldsymbol{r}_i, \boldsymbol{ct}_i; \text{q})$
21:      **end for**
22: **end if**

Algorithm 7 *HERS* Search

1: **Inputs:** $q \in \mathbb{R}^d$, $\mathcal{I}$, $\mathcal{D}$, database index k, ring dimension n ▷ unencrypted query and encrypted database

                                                      ▷ authentication at client

2: **for** $i = 1$ to d **do**
3:      $m \leftarrow BatchEncode(q_{i\cdot}; 0)$ ▷ i-th dim to all indices of plaintext
4:      $ct_i = Encrypt(m; \theta_{pk}, q, t)$
5: **end for**
6: Send $(\{ct_1, \ldots, ct_d\})$ to the server

                                                   ▷ authentication at server

7: $\mathcal{S} \leftarrow \emptyset$
8: **for** $v = 1$ to $\lceil \frac{k}{n} \rceil$ **do**
9:      $s \leftarrow Encrypt(0; \theta_{pk}, q, t)$ ▷ initialize all zero score ciphertext
10:      **for** $i = 1$ to d **do**
11:          $p \leftarrow CipherMultiply(ct_i, \mathcal{D}_i^v; w, l, q, t)$
12:          $s \leftarrow CipherAdd(s, p; q)$
13:      **end for**
14:      $\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$
15: **end for**
16: Send encrypted scores $\mathcal{S}$ back to client

                                                   ▷ authentication at client

17: $\mathcal{R} \leftarrow \emptyset$
18: **for** $l = 1$ to $\lceil \frac{k}{n} \rceil$ **do**
19:      $r \leftarrow Decrypt(\mathcal{S}_l; \theta_{sk}, q, t)$
20:      $\mathcal{R} \leftarrow \mathcal{R} \cup \{r\}$
21: **end for**
22: nearest neighbhor $\leftarrow \arg\max \mathcal{R}$

B.2   Search

Algorithm 7 describes our entire search procedure. This includes, query encryption, encrypted score computation, score decryption and argmax on the decrypted scores to find the nearest match.

C   Security Analysis

We adopt common assumptions in cryptography, i.e., the entities in our system (client and server) are *semi-honest* - each entity "follows the protocol properly with the exception that it keeps a record of all its intermediate computations"[33]. Under these assumptions, the security of *HERS* is built upon the security of the FV scheme, which in turn is based in the hardness of the *Ring Learning with Errors* problem [28]. Practically, this means that the security of our entire protocol hinges upon the fact that the ciphertext cannot be decrypted without access to the private (secret) decryption key which resides only on the client. The

encryption parameters (n, t, q) chosen by our experiments correspond to 128-bits of security.

The *HERS* system has three sources of vulnerability to attackers, namely, (1) the client device which holds the secret keys, (2) the communication channel between the client and server, and (3) the database which holds the encrypted representations and performs score computation in the encrypted domain. The physical and digital security of the client is most important in order to protect the secret keys. The security of the communication channel and the database server are guaranteed by the security of the FV scheme itself.

D   DeepMDS++

DeepMDS++ is comprised of repeating block units. A single block unit structure is shown in Table 6 and is comprised of two fully-connected layers.

Table 6: DeepMDS++ Block Unit (BU_i) Architecture

Layer Type	Input Dimensions	Output Dimensions
Fully Connected (Relu Activation)	I_D[1]	I_D
Fully Connected (No Activation)	I_D	O_D[2]

[1] I_D: Input dimension of current block unit.
[2] O_D: Output dimension of current block unit.

The number of block units in a given DeepMDS++ model varies, depending on the dimensionality of the original ambient space and the final intrinsic (or as close as possible to intrinsic without losing accuracy) space. The specific repeating block unit structures for the three representation models utilized in this paper are shown (assuming the lowest output dimension reported for each model in the paper) in Table 7 (Inception ResNet v2), Table 8 (ArcFace), and Table 9 (DeepPrint). When we report results for each of these three DeepMDS++ models at higher dimensions in the paper, we simply discard later block units and retrain. In all of our comparisons with the original DeepMDS [17], we utilize the same repeating block unit architectures for the DeepMDS baseline.

Table 7: DeepMDS++ (Inception ResNet v2)

Block Unit	Block Input Dimensions (I_D)	Block Output Dimensions (O_D)
BU_1	1,536	1,024
BU_2	1,024	512
BU_3	512	256
BU_4	256	128
BU_5	128	64
BU_6	64	32
BU_7	32	16

Table 8: DeepMDS++ (ArcFace)

Block Unit	Block Input Dimensions (I_D)	Block Output Dimensions (O_D)
BU_1	512	256
BU_2	256	128
BU_3	128	64

Table 9: DeepMDS++ (DeepPrint)

Block Unit	Block Input Dimensions (I_D)	Block Output Dimensions (O_D)
BU_1	192	128
BU_2	128	64
BU_3	64	32
BU_4	32	16