HERS: Homomorphically Encrypted Representation Search

Joshua J. Engelsma, *Student Member, IEEE,* Anil K. Jain, *Life Fellow, IEEE,*
and Vishnu Naresh Boddeti, *Member, IEEE*

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 with *minimal loss* of accuracy (ii) encrypting the compressed representation using the proposed fully homomorphic encryption scheme, and (iii) efficiently searching against a gallery of encrypted representations *directly in the encrypted domain, without decrypting them*. 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 (500 seconds; $275\times$ speed up over state-of-the-art for encrypted search against a gallery of 100 million). Code is available at https://github.com/human-analysis/hers-encrypted-image-search

Index Terms—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 encryption scheme like the Advanced Encryption Standard (AES) [1], 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.

• *Authors are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824.*
E-mail: {engelsm7, jain, vishnu}@msu.edu

1. https://bit.ly/2HD83Pq
2. https://wapo.st/39PQuaT
3. https://wapo.st/2V3kHPS
4. https://bit.ly/2OQhlM3

While much research continues in the area of improving the discriminative power of image representations (*e.g.* face recognition [2], [3] and image classification [4]), 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) [5] or (ii) violate user privacy by mining demographic attributes such as age, ethnicity, or gender [6]. More generally, it is well known that local features, such as deep CNN representations [7], SIFT [8], [9], HOG [10] and Bag-of-Visual-Words [11], can be inverted back into the image space with high fidelity.

A special class of encryption algorithms which enable basic arithmetic operations (multiplications and/or additions) in the encrypted domain are known as homomorphic encryption (HE) systems [12]. Since representations extracted by CNNs are typically compared using distance metrics that can be expressed in terms of additions and multiplications, like euclidean distance or cosine similarity, HE systems are a plausible solution to protect the representations within the database, and also during matching. HE systems are an attractive solution for secure feature matching systems. They have many desirable properties [13] including **diversity**, **revocability**, **security**, **performance**, and **privacy**. At the same time they do not suffer from any loss in matching performance since the encrypted computations are exact. Figure 1 shows a comparison of template matching in the unencrypted and the encrypted domain.

The key barrier against using HE schemes "off-the-shelf" [14] for protection of image representations is the computational complexity of arithmetic operations directly on the ciphertext (in the encrypted domain). This is espe-

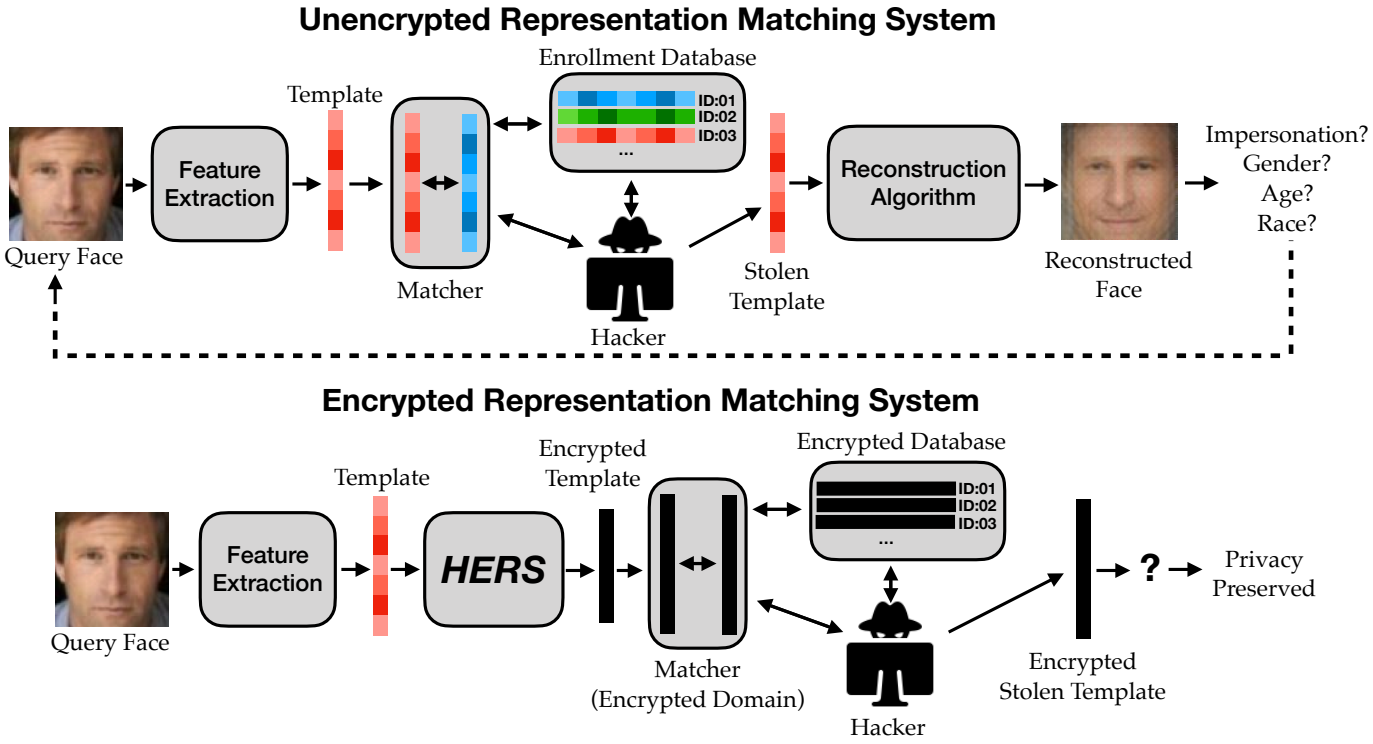

Fig. 1: **Overview:** Unencrypted and Encrypted face image search systems. Given an image representation (template), it is either (i) stored in a database (enrollment) or (ii) passed to the matcher for searching. Both the database and the matcher are potential points of attack. An attacker can either 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.

cially true of *fully*-homomorphic encryption schemes (FHE) which enable both additions *and* multiplications within the encrypted domain [14]. For example, the authors in [15] 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. The specific contributions of this paper are as follows,

1) 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 $11\times$ speed-up.
2) A dimensionality reduction scheme, dubbed DeepMDS++, based on DeepMDS [16], a state-of-the-art dimensionality reduction technique. For a 192-dim fingerprint representation from DeepPrint [17], this provides a $6\times$ speed-up at an only 0.1% loss in rank-1 search accuracy against a gallery of 100 million.
3) 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 $\approx 2\%$ of unencrypted accuracy) and timely (within 10 minutes) image search in the encrypted domain at scale (100 million gallery). The overall scheme provides a $186\times$ speed-up over a state-of-the-art $1:1$ matching of 32-dim encrypted feature vectors.

5. Single Instruction Multiple Data

2 Related Work

Privacy-Preserving Biometric Representations: Methods to secure representations of personally identifying information have been developed over the past decade [25]. These are summarized in [19] into two categories: (i) cryptographic protection, and (ii) pattern recognition based protection. Visual cryptography [26] is a common cryptographic approach for securing biometric data such as fingerprint [27] and face [28] images. Under some schemes, the visual perturbations need to be removed prior to performing a match, exposing the biometric data during authentication or

TABLE 1: Search Speed: 1 Million Gallery

Baseline Speed[1]	Baseline Processor Speed[†]
[18] 13,000 seconds[2]	2.4 GHz
[19] 10,000 seconds	3.3 GHz
[20] 12.7 Million seconds	3.3 GHz
[21] 500 seconds[3]	2.67 GHz
[22] 850,000 seconds	2.5 GHz
[23] 109,000 seconds	3.4 GHz
[24] 2 Million seconds	3.4 GHz

[†] HERS benchmarked on 3.3 GHz processor.
[1] HERS search speed against 1 million via 128-D templates is 300 seconds.
[2] Only 12-D features are used in [18].
[3] [21] uses Paillier Homomorphic encryption. It requires the client template to remain decrypted. HERS encrypts both client and gallery template at all times using FHE.

search. Fuzzy-vaults [29] are another cryptosystem which have been utilized for fingerprint [30] and iris [31] recognition. Pattern recognition based protection schemes have been proposed as an alternative to cryptosystems. Examples include non-invertible transformation functions [32] and cancelable biometrics [33], [34]. Additionally, key-binding systems have been proposed to merge a biometric template with a secret key [35], [36]. More recently, template security approaches based on neural networks [37] and representation geometry [38] have been designed for face representations. 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 very high levels of representation security with minimal loss in matching accuracy.

Privacy-Preserving Visual Recognition: Within the broader context of computer vision there is growing interest in privacy-preserving techniques. These approaches are based on cryptographic methods or computer vision techniques. Cryptographic methods include face detection [39], [40] using secure multi-party computation, image retrieval [41] by oblivious transfer (a building block of secure multi-party computation), face verification [42] using the Paillier Cryptosystem, video surveillance [43] using Secret Sharing, learning or inference over private data via differential privacy [44] and homomorphic encryption [45], [46], [47], [48], [49], [50]. Computer vision techniques include camera localization [51], [52] by lifting 2D and 3D points to 2D and 3D lines, embedding features into an adversarially designed subspace [53], detecting private computer screens [54] using CNNs, and activity recognition [55], [56] 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.

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 [12] most existing works utilizing HE are limited to binary templates and partial homomorphic encryption (PHE) (supports either encrypted additions or encrypted multiplications), or are computationally impractical [18], [20], [21], [22], [23], [24], [57], [58], [59], [60], [61]. From Table 1 it can be seen that existing methods are orders of magnitude slower than even the best existing baseline [19]. The only exception to this is the work described in [21], which uses Paillier Homomorphic encryption (requiring that the client template remain decrypted). Admittedly, many of the benchmarks reported in Table 1 are benchmarked on slower running CPUs than *HERS*. However, given the order of magnitude in time difference between *HERS* and the benchmarks, it is not plausible that a minor CPU upgrade will come close to bridging the gap in search speed. A few works [19], [62], [63] have demonstrated the effective use of Fully Homomorphic Encryption (FHE) schemes [14] (supports encrypted multiplications and additions). Cheon et al. [62] and Boddeti [19] 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. [17] 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 [19].

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 $m = 100,000,000$ gallery with representations that are of dimensionality 512 would still take over 38 hours and 9 TB memory with the improved FHE scheme proposed in [19]. 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.

Compressing Representations: A plethora of work has been done to estimate a low dimensional approximation of a data manifold which resides in a high dimensional space. In addition to the time tested classical methods for this task (*e.g.*, PCA, MDS, or Isomap), methods utilizing deep networks to learn more complex non-linear mappings have also been proposed. Some of these approaches combine representation learning together with dimensionality reduction via a primary learning objective (*e.g.* a classification loss) paired with a regularization loss term [64], [65]. In contrast, other approaches [16], [66], [67] train a completely separate dimensionality reduction algorithm on top of existing representations, such that access to the representation extraction network is not required. Since we want *HERS* to be extendable as a wrapper on top of any existing image search system, we pattern the dimensionality reduction component of *HERS* after DeepMDS [16], and significantly improve its performance.

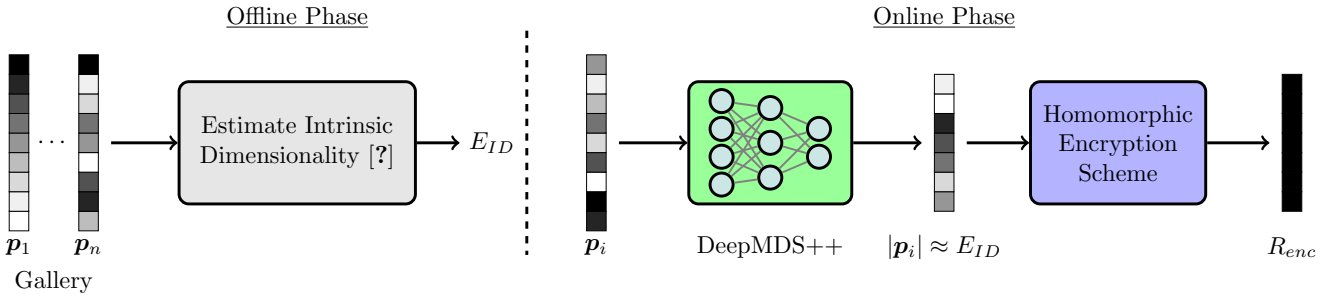

Fig. 2: Schematic Diagram of *HERS*. First, we estimate the intrinsic dimensionality of a given representation [16] in an offline stage from a gallery $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m$. Subsequently, we reduce the dimensionality of the representation towards its intrinsic dimensionality (E_{ID}) 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.

3 Approach

We posit that there are two avenues for improving 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 solutions 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 $\boldsymbol{q}$ and each template in $\boldsymbol{P}$. A common metric that is adopted for template matching is the Euclidean distance or the cosine similarity. At the core, both of these metrics involve matrix-vector products of the form: $r(\boldsymbol{P}, \boldsymbol{q}) = \boldsymbol{P}^T \boldsymbol{q}$. 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 $\boldsymbol{P}$ as well as the query $\boldsymbol{q}$ to prevent leakage of any private information. This can be achieved through a parameterized function that transforms a representation $\boldsymbol{z}$ from the original space into an alternate space, i.e., $\mathcal{E}(\boldsymbol{z}) = f(\boldsymbol{z}; \boldsymbol{\theta}_{pk})$, where $f(\cdot; \boldsymbol{\theta}_{pk})$ is the encryption function with public-key $\boldsymbol{\theta}_{pk}$ and $\boldsymbol{z} = g(\mathcal{E}(\boldsymbol{z}); \boldsymbol{\theta}_{sk})$, where $g(\cdot; \boldsymbol{\theta}_{sk})$ is the decryption function with private-key $\boldsymbol{\theta}_{sk}$. 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.,

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

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 or the query, without access to the private-key $\boldsymbol{\theta}_{sk}$ the attacker cannot reconstruct the underlying image or extract any other information inherent to the representation.

3.2 Fast and Secure Similarity Computation

We use the Fan-Vercauteren (FV) scheme [68] as our base FHE scheme[6]. 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 $R_t = \mathbb{Z}_t[x]/\boldsymbol{\Phi}_n(x)$, where $t \in \mathbb{Z}$ is an integer and $\boldsymbol{\Phi}_n(x)$ 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 $\boldsymbol{\theta}_{sk}$, (2) a public encryption key $\boldsymbol{\theta}_{pk}$, and (3) evaluation keys $\boldsymbol{\theta}_{ev}$ 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 $\boldsymbol{q} \in \mathbb{R}^d$ into an integer valued representation $\boldsymbol{q} \in \mathbb{Z}^d$. 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 [70] of the FV scheme. The primitives operate over an array of numbers instead of a single number, en-

6. A potentially more efficient but approximate alternative is CKKS [69], which can directly be applied to real-valued arithmetic.

TABLE 2: 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 [19]	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	$\mathcal{O}\left(dn\lceil \frac{m}{n} \rceil\right)$

coding multiple numbers within the same polynomial using the Chinese Remainder Theorem. The encoding scheme in Boddeti [19] is also based on the same principles but is specifically suitable for $1:1$ matching.

Given a query $\boldsymbol{q} \in \mathbb{Z}^d$ and a database of feature vectors $\boldsymbol{P} \in \mathbb{Z}^{d \times m}$, the encoding scheme in [19] encodes the feature vector of each sample (column) into a single polynomial. In contrast, in *HERS*, the client node encodes each dimension (row) of the representation into a separate polynomial. A query $\boldsymbol{q}$ is therefore encoded into d plaintexts as,

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

Similarly, for the database, each dimension of all the templates $\boldsymbol{P}_{i\cdot} \in \mathbb{Z}^m$ are encoded into polynomial, resulting in d plaintexts,

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

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

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

where the product $g * h$ is standard polynomial multiplication. As a consequence, the $m < n$ inner products can be computed through d polynomial multiplications[7]. 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 $\boldsymbol{p}$, 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.

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 $\mathcal{O}(n^2)$-fold increase[8] in computational complexity for addition and multiplication, respectively. Therefore, mitigating the number of ciphertext multiplications can lead to large gains in practical utility.

Table 2 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 [19] 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 $\sim \mathcal{O}\left(\frac{d}{n}\right)$.

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[9]). For example, we show a reduction of a DeepPrint [17] fingerprint representation from its 192-dim ambient space to a 32-dim space while

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

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

9. Intrinsic dimensionality [16] is the minimum number of dimensions needed to maintain the information present in the ambient representation.

losing only 0.1% rank-1 search accuracy, but getting a $\mathbf{6}\times$ search speed up within the encrypted domain.

There is growing evidence [16], [71] 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. Gong et al. [16] further learned a non-liner mapping, dubbed DeepMDS, to compress the representation from the ambient space to its intrinsic space.

Although DeepMDS [16] is able to learn demonstrably better mappings from the ambient to intrinsic space than existing dimensionality reduction schemes, it suffers from two major limitations which limits its direct utility in *HERS*. First, DeepMDS is limited in its generalization capability since it needs to be trained separately for each dataset. Furthermore, it was trained and evaluated on the same dataset. However, real-world application of *HERS* would necessitate a representation compression approach that generalizes across different datasets. Second, DeepMDS optimizes the average pair-wise loss of the samples in a mini-batch without explicitly accounting for the gross imbalance between the number of impostor and genuine pairs. Consequently, the mappings learned by DeepMDS do not focus sufficiently on preserving the local structure of the representation. Both of these drawbacks manifest themselves downstream in the form of loss in feature matching accuracy in the learned intrinsic space, especially when generalizing across different datasets. Therefore, DeepMDS does not serve our goal of compressing the representation while maintaining feature matching performance.

To mitigate the aforementioned limitations we propose **DeepMDS++**, a deep neural network based approach for compressing the representation. It is comprised of multiple non-linear layers, and is trained to map 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 $\mathbf{y} = f(\mathbf{x}; \boldsymbol{w})$

Let $\boldsymbol{G} \in \mathbb{R}^{b_1 \times d}$ and $\boldsymbol{I} \in \mathbb{R}^{b_2 \times d}$ be a set of genuine and imposter pair of templates in the ambient space for a mini-batch of size $b = b_1 + b_2$. Similarly, let $\hat{\boldsymbol{G}} \in \mathbb{R}^{b_1 \times s}$ and $\hat{\boldsymbol{I}} \in \mathbb{R}^{b_2 \times s}$ be the corresponding features in the intrinsic space. Given the pairwise similarities between genuine pairs $\boldsymbol{D}_G$ and imposter pairs $\boldsymbol{D}_I$ in the ambient space and in the intrinsic space ($\hat{\boldsymbol{D}}_G$, $\hat{\boldsymbol{D}}_I$), respectively, we optimize the distance between the similarities as follows,

$$\mathcal{L}_D = \frac{1}{b_1}\|\boldsymbol{D}_G - \hat{\boldsymbol{D}}_G\|_F^2 + \frac{1}{b_2}\|\boldsymbol{D}_I - \hat{\boldsymbol{D}}_I\|_F^2 \tag{5}$$

Covariance Penalty: To encourage generalization across datasets, and to explicitly discourage redundancy in DeepMDS++, we adopt a penalty on the covariance of the features. Our loss seeks to minimize the off-diagonal elements of the feature covariance matrix.

$$\mathcal{L}_c = \|\boldsymbol{C} - diag(\boldsymbol{C})\|_F^2 \tag{6}$$

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

Hard Pair Mining: To allow DeepMDS++ to focus on preserving the local structure of the features in the low-dimensional space, we introduce a hard-mining strategy [3]. This strategy first identifies "hard pairs" and include them in our mini-batch for training. In particular, we select those pairs for which the pairwise distance is not well preserved after their mapping into the intrinsic space. More formally, the mini-batch indices of the hard genuine pairs P_G and the hard imposter pairs P_I can be computed as:

$$P_G = argsort(\boldsymbol{D}_G - \hat{\boldsymbol{D}}_G) \quad P_I = argsort(\boldsymbol{D}_I - \hat{\boldsymbol{D}}_I) \tag{7}$$

where $argsort(\cdot)$ will return the indices of the rows in the features of genuine pairs $\mathbf{G}$ and the impostor pairs $\mathbf{I}$, with the highest error in pairwise distances following their mapping into the intrinsic space (i.e. the hardest pairs).

As we discuss next, experimental results demonstrate that incorporating these solutions aids DeepMDS++ in minimizing the loss of feature matching accuracy when compressing representations towards their intrinsic dimensionality, ultimately enabling us to perform matching in the encrypted domain more efficiently.

4 Experiments

We conduct numerical experiments to evaluate the different components of *HERS*, (i) the proposed encoding scheme that is tailored to $1 : m$ matching of features, (ii) DeepMDS++, and (iii) finally the combination the encoding scheme and DeepMDS++.

4.1 Implementation Details

DeepMDS++ is implemented in Tensorflow. Details of the network architecture are provided in Appendix C. The network is optimized with Adam [72] with an initial learning rate of 3×10^{-4} and with a weight decay of 4×10^{-5}. It is trained for 250 epochs using a NVIDIA GeForce RTX 2080 Ti GPU. For hard-pair mining, we used a mini-batch size of $N = 4,000$. 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 encrypted search in *HERS* is implemented using the SEAL library [73]. For all experiments, we used a 10-core Intel i9-7900X processor running at 3.30 GHz. All evaluations of the search were performed in 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[10] in SEAL i.e., a product of 3 very large primes, each 36 bits long.

10. In practice, much smaller values of q are sufficient for our purpose.

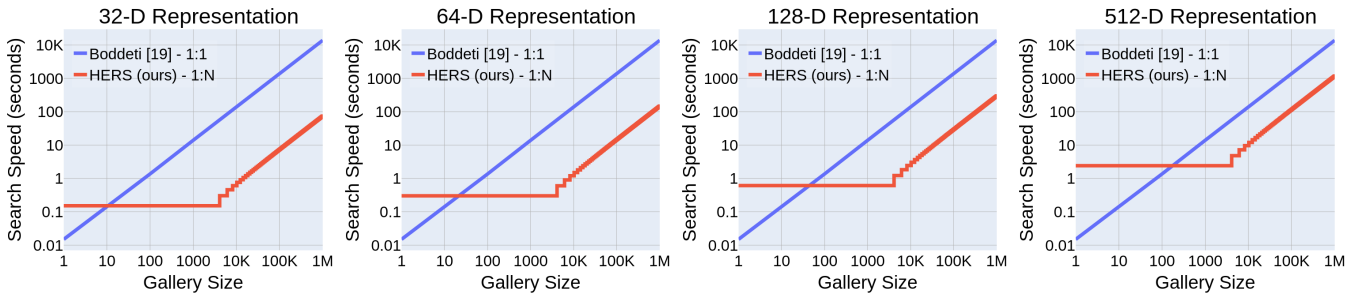

Fig. 3: Computational complexity comparison (log-log scale) of *HERS* with Boddeti [19] for $1 : m$ matching as a function of gallery size m and different representation dimensionality.

4.2 Evaluation Datasets

- **FaceScrub + MegaFace [74]:** MegaFace: 1 million distractor faces; FaceScrub: $\approx 3.5K$ celebrity faces. These are commonly coupled datasets for evaluating face search performance at scale.
- **CASIA [2]:** 450K face images from 10K subjects. 100K subset used to train DeepMDS++ prior to its application on MegaFace and FaceScrub, ensuring that our evaluation of DeepMDS++ is cross dataset.
- **NIST SD4 [75] + 100 Million Synthetic Background [76]:** NIST SD4: Contains 2,000 probe/mate inked-rolled fingerprint pairs. 100 Million Synthetic Fingerprints: 100 million synthetic fingerprint database used to evaluate scalability of *HERS*; a separate 100K fingerprints from [77] were used to train DeepMDS++.
- **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.

4.3 Representation Models

- **ArcFace [78]:** This model obtained state-of-the-art results on the MegaFace Challenge via (i) architectural refinements, (ii) a well curated training dataset, and (iii) an additive arc margin loss. We extracted 512-dim embeddings using a publicly available [11], pre-trained ArcFace model.
- **DeepPrint [17]:** This model is 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 by incorporating fingerprint domain knowledge into the training process.
- **Inception ResNet v2 [79]:** This model which combines inception modules with residual connections is one of the best models on ImageNet 2012 (Top-1 accuracy 80.3%). We use a pre-trained model[12] to extract ImageNet training and validation features (1,536-dim representations).

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

TABLE 3: *HERS*: Computational Complexity of Search Against 100 Million Gallery

Dimensions	32	64	128	512
Time (seconds)[1]	740	1,480	2,960	12,120
Memory (GB)	140	280	550	2,200

[1] 138,250 seconds and 9 TB at all dimensions for baseline (Boddeti [19]). (Using 10 parallel cores)

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 [19] , and (ii) we benchmark the matching and search accuracy at different representation dimensions against DeepMDS [16]. Finally, we (iiii) introduce a two stage encrypted search algorithm which combines *HERS* together with [19] to further improve the encrypted search speed at a scale of 100 million.

Efficiency: Figure 3, shows the computational complexity of the encryption component of *HERS* with [19]. Specifically, we consider representations at different dimensions (32-dim, 64-dim, 128-dim, and 512-dim) as we compress them towards their intrinsic dimensionality. This experiment highlights the synergistic benefit of coupling together encryption techniques with dimensionality reduction in order to perform encrypted search at scale. The results indicate that at small gallery sizes 1:1 matching from Boddeti [19] 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 280 GB for a gallery of size 100 million. In comparison [19] requires less memory for the probe at 19K, but exhausts 9 TB for the same gallery size. Table 3 reports the computational complexity of encrypted search, in terms of time and memory, against a gallery of 100 Million for different representation dimensionality. Since HERS is very amenable to parallel processing, we are able to search against 100 million 32-dim representations in 740 seconds ($186\times$ faster than [19]), with $64\times$ less memory when using 10 processes (cores) running in parallel. Finally, we note that returning the encrypted scores to the client requires transmitting m integers over the network. Even for a gallery

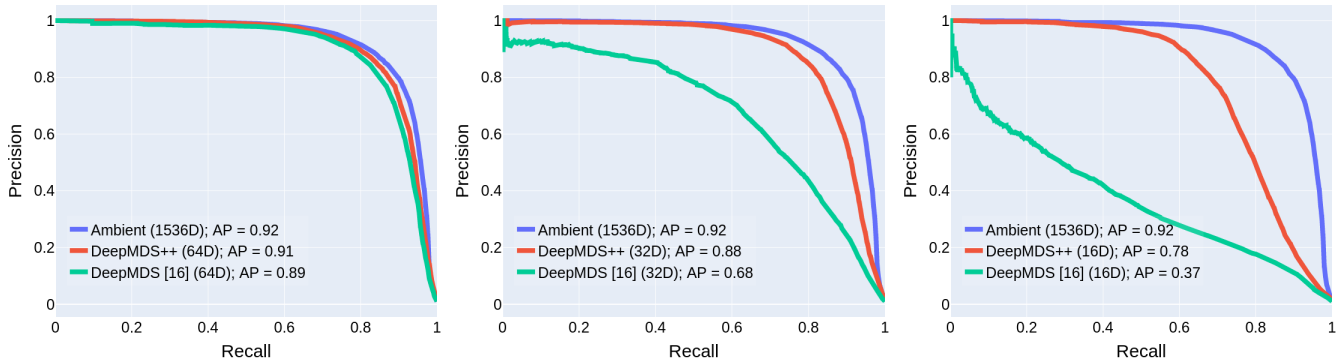

Fig. 4: Precision-Recall curves using Inception ResNet V2 on ImageNet 2012

size of 100 million, this only amounts to 100 MB of data. Decrypting the 100 million scores on the client device takes less than 1 second. The inference time for dimensionality reduction is 30 milliseconds. We obtained this inference time on an Intel Core i9-7900X CPU @ 3.30 GHz and 32 GB of RAM.

Accuracy: We evaluate the accuracy of DeepMDS++, the dimensionality reduction component of *HERS* in terms of: (i) Rank-1 Face Search Performance on MegaFace (1 million distractors), (ii) Rank-1 Fingerprint Search performance (using NIST SD4 against a gallery of 100 million synthetic fingerprints [76]), (iii) Top-1 classification accuracy and Precision-Recall curves from a subset of 100 classes of the ImageNet validation set, and (iv) Precision @ 10 (Precision in Top-10 retrieved samples) using the entire ImageNet validation set. 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.

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

(a) MegaFace (Gallery: 1 Million)

512D	256D	128D	64D
81.4	81.4	79.0	67.9

(b) Fingerprint (Gallery: 100 Million)

192D	64D	32D	16D
92.3	92.3	92.2	78.6

Table 4a and Table 4b show the Rank-1 accuracy of encrypted face and fingerprint 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) where the performance only drops by 0.1%. The 100 Million fingerprint gallery size indicates that DeepMDS++ is incredibly scalable. The degree of compression of the representation by DeepMDS++ closely mirrors the intrinsic dimensionality estimates[13] 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.

13. ID estimate code: https://github.com/human-analysis/intrinsic-dimensionality

TABLE 5: ImageNet
Average Precision (AP) and Top-1 Accuracy (%)

Dimension[1,2]	DeepMDS [16]		DeepMDS++ (ours)	
	AP	Top-1 (%)	AP	Top-1 (%)
128	0.92	85.4	0.92	**85.7**
64	0.89	83.4	**0.91**	**85.0**
32	0.68	66.1	**0.88**	**81.8**
16	0.37	42.7	**0.78**	**71.9**

[1] Original AP of 0.92 at 1536-dim
[2] Original Top-1 accuracy of 86.2% at 1536-dim

Table 5 reports 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 of 1536, 64, 32 and 16 for Inception ResNet v2 embeddings. We make the following observations from these results,

1) 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.
2) Representations compressed through DeepMDS++ are able to retain more discriminative information 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.
3) 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 [16].

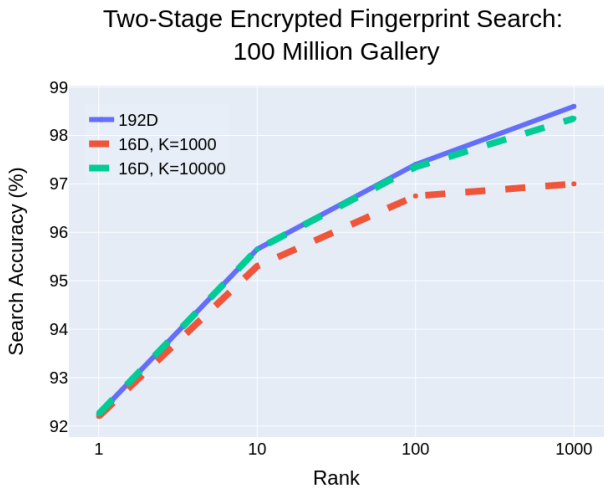

Fig. 5: DeepPrint rank-level search accuracy on NIST SD4 against a gallery of 100 million using our two-stage encrypted search. When $K = 10,000$ in the two-stage search, we can obtain nearly identical search performance as the original full 192-dim DeepPrint representation.

Two Stage Search: In Table 3, we demonstrated that *HERS* obtains encrypted search results against a gallery of 100 million in 740 seconds when the dimensionality of the representation is 32-dim. At 16-dim, this search time could be further reduced to 370 seconds. However, as shown in Table 4b, the fingerprint search accuracy deteriorates quite heavily when reducing the DeepPrint representation from 32-dim to 16-dim. To leverage the significant time savings from further dimensionality reduction to 16-dim, we propose a two stage encrypted search algorithm. In the first stage, we utilize the 16-dim DeepPrint representation as a "coarse matcher" in conjunction with *HERS* to find the top-K candidates. Then, we use the full 192-dim DeepPrint representation in conjunction with the encryption scheme from [19] to re-rank the top-K candidates. In this manner, we obtain the benefits of the 16-dim representation for faster search speeds, and by choosing an appropriate value for K, we can obtain nearly identical search accuracy to using the 192-dim representation for the full initial search. Note that we use [19] for the second stage of the search since it enables matching against the specific indices of the database returned by our first stage search. Figure 5 shows the rank-level search accuracy for DeepPrint on NIST SD4 against a gallery of 100 million synthetic fingerprints when (i) the initial 192-dim representation is used and (ii) the 16-dim representation is used as a first stage search algorithm, and then the 192-dim representation is used to re-rank. From these results, we observe that when $K = 10,000$ for the two-stage search, we can match the accuracy of the full 192-dim representation. This in turn enables cutting the encrypted search time against 100 million fingerprints from 4,500 seconds (at 192-dim) to 500 seconds (370 seconds for stage-1 search, 130 seconds for stage-2 re-ranking) without loss of accuracy.

Admittedly, one downside of the two stage encrypted search scheme is that it requires encrypting the representations twice (once with the *HERS* encoding and once with the encoding from [19]). This results in the overhead of the 9 TB gallery of [19] (for 100 million representations). In general, when designing an encrypted search system, we are looking to optimize along axes of speed, memory, accuracy, and security. Depending on the demands of the particular application, the point at which we land along these axes can be shifted. *HERS* explores different points along these axes via dimensionality reduction of features (small amounts of accuracy for significant speed savings), and a data-encoding scheme (sacrificing speed and memory at smaller gallery sizes for gains in both at larger galleries). Likewise, the two-stage search schema trades off memory overhead for encrypted search speed and accuracy gains without compromising security.

Ablation: Finally, we conduct an ablation on DeepMDS++. Table 6 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.

TABLE 6: Precision @ Rank 10 on ImageNet ILSVRC-2012 Validation Set (%)

Dim.[1]	**DeepMDS++** (proposed)	w/o Hard Mining	w/o $\mathcal{L}_c$ w/o Hard Mining (DeepMDS [16])
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%.

5 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" [80]. 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 [81]. 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. An attractive property of the FHE scheme [68] used in this paper is that it offers post-quantum security for an appropriate choice of encryption parameters as outlined in the homomorphic encryption standard [82]. The encryption parameters (n, t, q) chosen by our experiments correspond to 128-bits of post-quantum 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. Indeed, if the secret key were obtained from the client, the encrypted templates could be inverted and the templates exposed. Thus the security of the client is of utmost importance. We also note that in HERS, it is possible that some additional protection is rendered to the templates via our dimensionality reduction algorithm (compact feature vectors may be harder to reconstruct useful information from than the original full length representations). The security of the communication channel and the database server are guaranteed by the security of the FV scheme itself.

Score Inversion: We now consider an attack based on access to the decrypted scores. We posit that access to the decrypted score without access to the unencrypted features does not allow an attacker to estimate the user's biometric signature. From an adversary's perspective, recovering the unknown feature $\boldsymbol{q} \in \mathbb{R}^d$ from the scores $\boldsymbol{r} = \boldsymbol{P}^T\boldsymbol{q}$ can be expressed as an optimization problem,

$$\begin{aligned} \min_{\boldsymbol{q}} \quad & \|\boldsymbol{r} - \boldsymbol{P}^T\boldsymbol{q}\|_2^2 \\ s.t. \quad & \boldsymbol{q}^T\boldsymbol{q} = 1 \end{aligned} \tag{8}$$

where we assume that the unknown feature is normalized. If the feature is not normalized, the constraint can be removed. The solution of this optimization problem is, $\boldsymbol{q} = \left(\boldsymbol{P}\boldsymbol{P}^T + \lambda \boldsymbol{I}\right)^{-1}\boldsymbol{P}\boldsymbol{r}$. Observe that the solution depends on having access to the raw features $\boldsymbol{P}$ in the database. However, without access to the secret decryption keys, the database $\boldsymbol{P}$ is available only in the encrypted form. Therefore, even with access to the unencrypted scores, the unknown feature cannot be recovered.

6 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 [83]): 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 [84]: 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 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*[14]. 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 as discussed in Section 5. 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.

7 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 10 minutes) and accurate (within $\approx 2\%$ of unencrypted accuracy) image search (for face, fingerprint and ImageNet) against 100 Million gallery in the encrypted domain.

8 Acknowledgements

This research was supported in part by a grant from The NSF Center for Identification Technology Research (CITeR). Vishnu Boddeti was supported in part by award 60NANB18D210 from U.S. Department of Commerce, National Institute of Standards and Technology.

References

[1] S. Heron, "Advanced encryption standard (AES)," *Network Security*, vol. 2009, no. 12, pp. 8–12, 2009.
[2] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.
[3] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[5] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the reconstruction of face images from deep face templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1188–1202, 2018.
[6] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *IEEE International Conference on Computer Vision*, 2015.
[7] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
[8] P. Weinzaepfel, H. Jégou, and P. Pérez, "Reconstructing an image from its local descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

14. Approximate comparison between homomorphically encrypted numbers is now supported by the CKKS FHE scheme [85]. In principle, although this capability can be leveraged to compute the *argmax* of the encrypted scores, it is computationally too prohibitive and is a topic of future research.

[9] F. Pittaluga, S. J. Koppal, S. B. Kang, and S. N. Sinha, "Revealing scenes by inverting structure from motion reconstructions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: Visualizing object detection features," in *IEEE International Conference on Computer Vision*, 2013.

[11] H. Kato and T. Harada, "Image reconstruction from bag-of-visual-words," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[12] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge university press, 2009.

[13] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Springer Science & Business Media, 2009.

[14] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 2011.

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

[16] S. Gong, V. N. Boddeti, and A. K. Jain, "On the intrinsic dimensionality of image representations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[17] J. J. Engelsma, K. Cao, and A. K. Jain, "Learning a fixed-length fingerprint representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[18] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *International Conference on Information Security and Cryptology*, pp. 229–244, Springer, 2009.

[19] V. N. Boddeti, "Secure face matching using fully homomorphic encryption," in *IEEE Conference on Biometrics Theory, Applications and Systems*, 2018.

[20] J. R. Troncoso-Pastoriza and F. Pérez-González, "Fully homomorphic faces," in *2012 19th IEEE International Conference on Image Processing*, pp. 2657–2660, IEEE, 2012.

[21] M. Gomez-Barrero, E. Maiorana, J. Galbally, P. Campisi, and J. Fierrez, "Multi-biometric template protection based on homomorphic encryption," *Pattern Recognition*, vol. 67, pp. 149–163, 2017.

[22] P. Drozdowski, N. Buchmann, C. Rathgeb, M. Margraf, and C. Busch, "On the application of homomorphic encryption to face identification," in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, 2019.

[23] T. Kim, Y. Oh, and H. Kim, "Efficient privacy-preserving fingerprint-based authentication system using fully homomorphic encryption," *Security and Communication Networks*, vol. 2020, 2020.

[24] T. Yang, Y. Zhang, J. Sun, and X. Wang, "Privacy enhanced cloud-based facial recognition," *Neural Processing Letters*, pp. 1–9, 2021.

[25] C. Rathgeb and A. Uhl, "A survey on biometric cryptosystems and cancelable biometrics," *EURASIP Journal on Information Security*, vol. 2011, no. 1, pp. 1–25, 2011.

[26] M. Naor and A. Shamir, "Visual cryptography," in *Workshop on the Theory and Application of of Cryptographic Techniques*, 1994.

[27] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, "Biometric encryption," in *ICSA Guide to Cryptography*, vol. 22, McGraw-Hill, 1999.

[28] A. Ross and A. Othman, "Visual cryptography for biometric privacy," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 70–81, 2010.

[29] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[30] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints," in *International Conference on Audio-and Video-Based Biometric Person Authentication*, pp. 310–319, 2005.

[31] Y. J. Lee, K. R. Park, S. J. Lee, K. Bae, and J. Kim, "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)*, vol. 38, no. 5, pp. 1302–1313, 2008.

[32] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[33] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 54–65, 2015.

[34] Z. Jin, J. Y. Hwang, Y.-L. Lai, S. Kim, and A. B. J. Teoh, "Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 393–407, 2017.

[35] V. N. Boddeti and B. V. Kumar, "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*, vol. 35, no. 9, pp. 2064–2077, 2012.

[36] K. Nandakumar, A. Nagar, and A. K. Jain, "Hardening fingerprint fuzzy vault using password," in *International Conference on Biometrics*, 2007.

[37] G. Mai, K. Cao, X. Lan, and P. C. Yuen, "Secureface: Face template protection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 262–277, 2020.

[38] S. Kim, Y. Jeong, J. Kim, J. Kim, H. T. Lee, and J. H. Seo, "Ironmask: Modular architecture for protecting deep face template," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[39] S. Avidan and M. Butman, "Blind vision," in *European Conference on Computer Vision*, 2006.

[40] S. Avidan and M. Butman, "Efficient methods for privacy preserving face detection," in *Advances in Neural Information Processing Systems*, 2007.

[41] J. Shashank, P. Kowshik, K. Srinathan, and C. Jawahar, "Private content based image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[42] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Blind authentication: a secure crypto-biometric verification protocol," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 255–268, 2010.

[43] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Efficient privacy preserving video surveillance," in *International Conference on Computer Vision*, 2009.

[44] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

[45] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *International Conference on Machine Learning*, 2016.

[46] R. Yonetani, V. Naresh Boddeti, K. M. Kitani, and Y. Sato, "Privacy-preserving visual learning using doubly permuted homomorphic encryption," in *IEEE International Conference on Computer Vision*, 2017.

[47] A. Brutzkus, R. Gilad-Bachrach, and O. Elisha, "Low latency privacy preserving inference," in *International Conference on Machine Learning*, pp. 812–821, PMLR, 2019.

[48] N. Jain, K. Nandakumar, N. Ratha, S. Pankanti, and U. Kumar, "Efficient cnn building blocks for encrypted data," *arXiv preprint arXiv:2102.00319*, 2021.

[49] K. Sarpatwar, K. Nandakumar, N. Ratha, J. Rayfield, K. Shanmugam, S. Pankanti, and R. Vaculin, "Efficient encrypted inference on ensembles of decision trees," *arXiv preprint arXiv:2103.03411*, 2021.

[50] K. Nandakumar, N. Ratha, S. Pankanti, and S. Halevi, "Towards deep neural network training on encrypted data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[51] P. Speciale, J. L. Schonberger, S. B. Kang, S. N. Sinha, and M. Pollefeys, "Privacy preserving image-based localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[52] P. Speciale, J. L. Schonberger, S. N. Sinha, and M. Pollefeys, "Privacy preserving image queries for camera localization," in *IEEE International Conference on Computer Vision*, 2019.

[53] M. Dusmanu, J. L. Schönberger, S. N. Sinha, and M. Pollefeys, "Privacy-preserving visual feature descriptors through adversarial affine subspace embedding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[54] M. Korayem, R. Templeman, D. Chen, D. Crandall, and A. Kapadia, "Enhancing lifelogging privacy by detecting screens," in *CHI Conference on Human Factors in Computing Systems*, 2016.

[55] Z. Ren, Y. Jae Lee, and M. S. Ryoo, "Learning to anonymize faces for privacy preserving action detection," in *European Conference on Computer Vision*, 2018.

[56] M. S. Ryoo, B. Rothrock, C. Fleming, and H. J. Yang, "Privacy-preserving human activity recognition from extreme low resolution," in *AAAI Conference on Artificial Intelligence*, 2017.

[57] M. Barni, G. Droandi, and R. Lazzeretti, "Privacy protection in biometric-based recognition systems: A marriage between cryptography and signal processing," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 66–76, 2015.

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

[59] M. Gomez-Barrero, E. Maiorana, J. Galbally, P. Campisi, and J. Fierrez, "Multi-biometric template protection based on homomorphic encryption," *Pattern Recognition*, vol. 67, pp. 149–163, 2017.

[60] C. Xiang, C. Tang, Y. Cai, and Q. Xu, "Privacy-preserving face recognition with outsourced computation," *Soft Computing*, vol. 20, no. 9, pp. 3735–3744, 2016.

[61] A. E. C. Asuncion, B. C. T. Guadalupe, and W. E. S. Yu, "Implementation and analysis of homomorphic facial image encryption and manipulation," in *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, pp. 158–166, 2019.

[62] J. H. Cheon, H. Chung, M. Kim, and K.-W. Lee, "Ghostshell: Secure biometric authentication using integrity-based homomorphic evaluations.," *IACR Cryptology ePrint Archive*, vol. 2016, p. 484, 2016.

[63] J. Kolberg, P. Drozdowski, M. Gomez-Barrero, C. Rathgeb, and C. Busch, "Efficiency analysis of post-quantum-secure face template protection schemes based on homomorphic encryption," in *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–4, IEEE, 2020.

[64] A. Sanyal, V. Kanade, P. H. Torr, and P. K. Dokania, "Robustness via deep low-rank representations," *arXiv preprint arXiv:1804.07090*, 2018.

[65] J. Lezama, Q. Qiu, P. Musé, and G. Sapiro, "Ole: Orthogonal low-rank embedding-a plug and play geometric loss for deep learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8109–8118, 2018.

[66] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[67] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1735–1742, IEEE, 2006.

[68] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption.," *IACR Cryptology ePrint Archive*, vol. 2012, p. 144, 2012.

[69] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 409–437, Springer, 2017.

[70] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "$GAZELLE$: A low latency framework for secure neural network inference," in *USENIX Security Symposium)*, pp. 1651–1669, 2018.

[71] A. Ansuini, A. Laio, J. H. Macke, and D. Zoccolan, "Intrinsic dimension of data representations in deep neural networks," in *Advances in Neural Information Processing Systems*, 2019.

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[73] "Microsoft SEAL (release 3.4)." https://github.com/Microsoft/SEAL, Oct. 2019. Microsoft Research, Redmond, WA.

[74] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[75] C. I. Watson and C. Wilson, "Nist special database 4," *Fingerprint Database, National Institute of Standards and Technology*, vol. 17, no. 77, 1992.

[76] V. Mistry, J. J. Engelsma, and A. K. Jain, "Fingerprint synthesis: Search with 100 million prints," in *International Joint Conference on Biometrics*, 2020.

[77] S. Yoon and A. K. Jain, "Longitudinal study of fingerprint recognition," *Proceedings of the National Academy of Sciences*, vol. 112, no. 28, pp. 8555–8560, 2015.

[78] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[79] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI Conference on Artificial Intelligence*, 2017.

[80] Oded Goldreich, *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[81] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 2010.

[82] M. R. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. E. Lauter, *et al.*, "Homomorphic encryption standard.," *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 939, 2019.

[83] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, 1999.

[84] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, vol. 78, 1998.

[85] J. H. Cheon, D. Kim, and D. Kim, "Efficient homomorphic comparison methods with optimal complexity," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 221–256, Springer, 2020.

Joshua J. Engelsma graduated magna cum laude with a B.S. degree in computer science from Grand Valley State University, Allendale, Michigan, in 2016. He obtained a PhD degree in Computer Science and Engineering at Michigan State University, May 2021. His research interests include pattern recognition, computer vision, and image processing with applications in biometrics. He won the best paper award at the 2019 IEEE International Conference on Biometrics (ICB), and the 2020 Michigan State University College of Engineering Fitch Beach Award.

Anil K. Jain is a University Distinguished Professor in the Department of Computer Science at Michigan State University. He is a Fellow of the ACM, IEEE, IAPR, AAAS and SPIE. His research interests include pattern recognition and biometric authentication. He served as the editor-in-chief of the IEEE Transactions on Pattern Analysis and Machine Intelligence, a member of the United States Defense Science Board and the Forensics Science Standards Board. He has received Fulbright, Guggenheim, Alexander von Humboldt, and IAPR King Sun Fu awards. He is a member of the United States National Academy of Engineering, a member of The World Academy of Science, and foreign members of the Indian National Academy of Engineering and the Chinese Academy of Sciences.

Vishnu Naresh Boddeti is an Assistant Professor in the computer science department at Michigan State University. He received a Ph.D. degree in Electrical and Computer Engineering program at Carnegie Mellon University in 2013. His research interests are in Computer Vision, Pattern Recognition and Machine Learning. He received the best paper award at BTAS 2013, the best student paper award at ACCV 2018, and the best paper award at GECCO 2019.

In this Appendix, we include, (a) details of the base homomorphic encryption scheme [68] in Section A, (b) detailed algorithms for the enrollment and search phase of *HERS* in Section B, (c) security analysis of *HERS* in Section 5, and (d) architectural details of DeepMDS++ in Section C.

APPENDIX A
FULLY HOMOMORPHIC ENCRYPTION

For completeness, we describe the Fan-Vercauteren [68] 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 $\boldsymbol{m}$ and the ciphertext polynomial as $\boldsymbol{ct}$.

Fan-Vercauteren Scheme [68]: 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

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

Algorithm 2 Encryption

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

Algorithm 3 Decryption

procedure DECRYPT($\boldsymbol{ct}$, $\boldsymbol{\theta}_{sk}$, q, t)
  $\boldsymbol{pt} = \left[\left\lfloor \frac{t}{q} [\boldsymbol{ct}[0] + \boldsymbol{ct}[1]\boldsymbol{\theta}_{sk}]_q \right\rceil\right]_t$
  return $\boldsymbol{pt}$
end procedure

Algorithm 4 Ciphertext Addition

procedure CIPHERADD($\boldsymbol{ct}_0$, $\boldsymbol{ct}_1$, q)
  $\boldsymbol{ot} = ([\boldsymbol{ct}_0[0] + \boldsymbol{ct}_1[0]]_q, [\boldsymbol{ct}_0[1] + \boldsymbol{ct}_1[1]]_q)$
  return $\boldsymbol{ot}$
end procedure

Algorithm 5 Ciphertext Multiplication

procedure CIPHERMULTIPLY($\boldsymbol{ct}_0$, $\boldsymbol{ct}_1$, w, l, q, t)
  $\boldsymbol{c}_0 = \left[\left\lfloor \frac{t}{q}(\boldsymbol{ct}_0[0]\boldsymbol{ct}_1[0]) \right\rceil\right]_q$
  $\boldsymbol{c}_1 = \left[\left\lfloor \frac{t}{q}(\boldsymbol{ct}_0[0]\boldsymbol{ct}_1[1] + \boldsymbol{ct}_0[1]\boldsymbol{ct}_1[0]) \right\rceil\right]_q$
  $\boldsymbol{c}_2 = \left[\left\lfloor \frac{t}{q}(\boldsymbol{ct}_0[1]\boldsymbol{ct}_1[1]) \right\rceil\right]_q$
  $\boldsymbol{c}_2 = \sum_{i=0}^{l} c_2^{(i)} w^i$ and compute
  $\boldsymbol{c}'_0 = \boldsymbol{c}_0 + \sum_{i=1}^{l} \boldsymbol{\theta}_{ev}[i][0]c_2^{(i)}$
  $\boldsymbol{c}'_1 = \boldsymbol{c}_1 + \sum_{i=1}^{l} \boldsymbol{\theta}_{ev}[i][1]c_2^{(i)}$
  return $(\boldsymbol{c}'_0, \boldsymbol{c}'_1)$
end procedure

APPENDIX 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: $\boldsymbol{ct}_i = Encrypt(\boldsymbol{m}; \boldsymbol{\theta}_{pk}$,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: $\boldsymbol{r}_i \leftarrow Encrypt(\mathbf{0}; \boldsymbol{\theta}_{pk}$,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: $\boldsymbol{r}_i \leftarrow CipherAdd(\boldsymbol{r}_i, \boldsymbol{ct}_i$;q)
21: **end for**
22: **end if**

Algorithm 7 *HERS* Search

1: **Inputs:** $\boldsymbol{q} \in \mathbb{R}^d$, $\mathcal{I}$, $\mathcal{D}$, database index k, ring dimension n ▷ unencrypted query and encrypted database ▷ authentication at client
2: **for** $i = 1$ to d **do**
3: $\boldsymbol{m} \leftarrow BatchEncode(\boldsymbol{q}_{i\cdot}; 0)$ ▷ i-th dim to all indices of plaintext
4: $\boldsymbol{ct}_i = Encrypt(\boldsymbol{m}; \boldsymbol{\theta}_{pk}$,q,t)
5: **end for**
6: Send $(\{\boldsymbol{ct}_1,\ldots,\boldsymbol{ct}_d\})$ to the server ▷ authentication at server
7: $\mathcal{S} \leftarrow \emptyset$
8: **for** $v = 1$ to $\lceil \frac{k}{n} \rceil$ **do**
9: $\boldsymbol{s} \leftarrow Encrypt(\mathbf{0}; \boldsymbol{\theta}_{pk}$,q,t) ▷ initialize all zero score ciphertext
10: **for** $i = 1$ to d **do**
11: $\boldsymbol{p} \leftarrow CipherMultiply(\boldsymbol{ct}_i, \mathcal{D}_i^v$;w,l,q,t)
12: $\boldsymbol{s} \leftarrow CipherAdd(\boldsymbol{s}, \boldsymbol{p}$;q)
13: **end for**
14: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{s}\}$
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: $\boldsymbol{r} \leftarrow Decrypt(\mathcal{S}_l; \boldsymbol{\theta}_{sk}$,q,t)
20: $\mathcal{R} \leftarrow \mathcal{R} \cup \{\boldsymbol{r}\}$
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.

APPENDIX C
DEEPMDS++

DeepMDS++ is comprised of repeating block units. A single block unit structure is shown in Table 7 and is comprised of two fully-connected layers.

TABLE 7: 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 8 (Inception ResNet v2), Table 9 (ArcFace), and Table 10 (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 [16], we utilize the same repeating block unit architectures for the DeepMDS baseline.

TABLE 8: 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 9: 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 10: 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