Multi Random Projection Inner Product Encryption, Applications to Proximity Searchable Encryption for the Iris Biometric[?,??]

Chloe Cachet[a,], Sohaib Ahmad[a], Luke Demarest[a], Serena Riback[a,c], Ariel Hamlin[b], Benjamin Fuller[a]

[a]University of Connecticut, 371 Faireld Way, Storrs, CT 06269, United States.
[b]MIT Lincoln Laboratory, 244 Wood Street Lexington, MA 02420, United States.
[c]Publicis Sapient, 40 Water Street, Boston, MA 02109, United States.

Abstract

Biometric databases collect people's information and allow users to perform proximity searches (nding all records within a bounded distance of the query point) with few cryptographic protections. This work studies proximity searchable encryption applied to the iris biometric.

Prior work proposed inner product functional encryption as a technique to build proximity biometric databases (Kim et al., SCN 2018). This is because binary Hamming distance is computable using an inner product. This work identies and closes two gaps in using inner product encryption for biometric search:

1. Biometrics naturally use long vectors often with thousands of bits. Many inner product encryption schemes generate a random matrix whose dimension scales with vector size and have to invert this matrix. As a result, setup is not feasible on commodity hardware unless we reduce the dimension of the vectors. We explore state-of-the-art techniques to reduce the dimension of the iris biometric and show that all known techniques harm the accuracy of the resulting system. That is, for small vector sizes multiple unrelated biometrics are returned in the search. For length 64 vectors,

---

at a 90% probability of the searched biometric being returned, 10% of stored records are erroneously returned on average.

Rather than changing the feature extractor, we introduce a new cryptographic technique that allows one to generate several smaller matrices. For vectors of length 1024 this reduces the time to run setup from 23 days to 4 minutes. At this vector length, for the same 90% probability of the searched biometric being returned, :02% of stored records are erroneously returned on average.

2. Prior inner product approaches leak distance between the query and all stored records. We refer to these as distance-revealing. We show a natural construction from function hiding, secret-key, predicate, inner product encryption (Shen, Shi, and Waters, TCC 2009). Our construction only leaks access patterns and which returned records are the same distance from the query. We refer to this scheme as distance-hiding.

We implement and benchmark one distance-revealing and one distance-hiding scheme. The distance-revealing scheme can search a small (hundreds) database in 4 minutes while the distance-hiding scheme is not yet practical, requiring 3:5 hours.

As a technical contribution of independent interest, we show that our scheme can be instantiated using symmetric pairing groups reducing the cost of search by roughly a factor of three. We believe this analysis extends to other schemes based on projections to a random linear map and its inverse analyzed in the generic group model.

Keywords: Searchable encryption, biometrics, proximity search, inner product encryption.

---

1. Introduction

Biometrics are measurements of physical phenomena of the human body. We focus on the iris biometric in this work. The iris is an interesting biometric because it has high entropy, stays stable throughout life, is not determined genetically, and is easily accessible [2]. Iris data, like all biometric data, is noisy, which means that two readings from the same iris are unlikely to be identical. Feature extractors convert such physical phenomena to a digital representation that is more stable but still noisy. The output of feature extractors is called a template. Biometric databases are used for both security critical applications (such as access control) and privacy critical applications (such as immigration). Let D be some distance metric and t be some distance threshold. Applications building on biometric templates require:

1. Low False Reject Rate (FRR) templates from the same biometric are within distance t with high probability, and
2. Low False Accept Rate (FAR) templates from two dierent biometrics are within distance t with low probability.

Learning stored biometric templates enables an attacker to reverse this value into a convincing biometric [3, 4, 5], enabling presentation attacks [6, 7, 8] that can compromise users' accounts and devices. Since biometrics cannot be updated, such a compromise lasts a lifetime.

Securing biometric data is not straightforward. Using plain encryption or hashes (as for passwords storage) would eectively protect compromised templates but also prevents the server from performing any distance comparison. Another option is for the server to hold a key to decrypt the data either in plaintext or inside a trusted execution environment (TEE). Attacks show protecting such information even inside a TEE is dicult [9, 10, 11].

Searchable encryption [12, 13, 14, 15] enables servers to be queried without decrypting the data. For a distance metric D, proximity searchable encryption returns all records that are within distance t. That is, for a dataset $x_1, \ldots, x_\ell$ and a query y, one should return all x_i such that $D(x_i, y) \le t$. Since biometric data is inherently noisy, proximity searchable encryption is a key tool to secure biometric databases while allowing queries.

Iris feature extractors usually produce binary vectors that are similar in Hamming distance[1] (ngerprints are usually compared by set dierence, faces with L2 norm). Kim et al. proposed to use secret-key, function-hiding inner product encryption or $IPE_{fh,sk}$ for encrypted comparison of binary Hamming bio-metrics [16, 17]. $IPE_{fh,sk}$ allows computation of inner product without revealing underlying values. Inner product of vectors x, y in $\{-1, 1\}^n$ encodes Hamming distance:

$$D(x, y) = (n - \langle x, y \rangle)/2.$$

More formally the functionality of $IPE_{fh,sk}$ is as follows: as in standard encryption, one generates a secret key $sk \leftarrow Setup()$ and a ciphertext $ct_x \leftarrow Encrypt(x, sk)$. Then one can generate a token $tk_y \leftarrow TokGen(y, sk)$, and use this token in Decrypt (without sk) to learn the inner product $\langle x, y \rangle$. That is,

$$Decrypt(ct_x, tk_y) = \langle x, y \rangle.$$

One can use $IPE_{fh,sk}$ to build proximity search by encrypting $ct_i \leftarrow Encrypt(x_i, sk)$ and providing all ct_i to the database server (additional data can be associated with x_i using traditional encryption). For queries y the client provides $tk_y \leftarrow TokGen(y, sk)$ to the server. The server can compute the inner product between the query and each stored record and should return all records with the appropriate inner product. The server does work proportional to the database size. This is in contrast to keyword searchable encryption where the desiderata is for the server's work to be proportional to the result set size.

We identify and close two gaps in the use of inner product encryption to build proximity searchable encryption for the iris.

[1]Note that real-valued vectors for the Euclidean distance can be converted to binary vectors for the Hamming distance using mean or median thresholding, where values above the mean/median are encoded as 1 and values below as 0.

1.1. Our Contribution

Multi Random Projection Inner Product Encryption. Daugman's seminal iris feature extractor [18, 19] produces a vector of length n = 1024, the open source OSIRIS [20] system uses n = 32768 by default, and recent neural network feature extractors [21] use n = 2048.

The most ecient $IPE_{fh;sk}$ schemes rely on dual pairing vector spaces [22] in bilinear groups. The secret key for such constructions is a random matrix A $2 \ F_q^{n \ n}$ and its inverse A ^{1}; q is a large prime that is the order of the bilinear pairing. Setup for the scheme must invert a random A $2 \ F_q^{n \ n}$.

This operation is prohibitive for n > 1000, as is the case for iris feature extractors. For the most ecient known scheme which we call Random Projection with Check or RProjC [16], the authors' parallel implementation of key generation in FLINT [23] (on a modern 16 core machine), generating keys for n = 240, took 4:6 hours. In our experiments, Setup time grows cubicly as expected.[2] Through interpolation, we estimate the time to generate keys for n = 1024 at 23 days.

While one can train feature extractors with smaller n, we show (in Section 4) that known techniques harm the quality of the biometric features, making the irises of dierent people appear similar. The false accept vs false reject rate tradeo degrades, leaving the application with the choice of either not matching readings of the same iris or matching readings of dierence individuals' irises. Both choices have consequences for the resulting application.

In Section 4.1 and Table 3, we show that for a small size dataset of 356 individuals using a feature extractor with n = 64 and a distance t that enables a 90% true accept rate, searching for an individual in the dataset returns 40 incorrect biometrics with an average query! By comparison when n = 1024, queries return :06 incorrect biometrics on average. Datasets with more individuals are not available; we expect this rate to be consistent across dataset sizes.

Section 5 introduces a new transform for inner product encryption that generates multiple matrices $A_1; ::::; A$ and their inverses during key generation where each A_i is an (N + 1) (N + 1) matrix, where N = dn=e, instead of a single large pair A; A ^{1}. Recall that the encryption and token generation algorithms take as input vectors x and y respectively. Vectors x and y are then split into component vectors of size N. Correctness then relies on the fact that

$$hx; yi = \sum_{i=0}^{1} hx_i; y_i i$$

where $x_i; y_i$ are component vectors of x and y. To hide partial information, both x and y are augmented when they are split into component vectors:

$$x_i^C = 1 \ jj \ x_{iN} ; ::::; x_{iN+(N \ 1)} y_i$$
$$^C = _i \ jj \ y_{iN} ; ::::; y_{iN+(N \ 1)}$$

---

[2] We have not evaluated sub-cubic matrix inversion in nite elds.

for $i = 0, \ldots, \ell-1$ and $\lambda_0, \ldots, \lambda_{\ell-1}$ is a linear secret sharing of 0 that is chosen in TokGen. The intuition is that any collection of $\ell-1$ or fewer components represents a random group element, so one cannot learn information about inner products between vector components. We show security of a prior IPE scheme with multi random projection in Section 5 (we also apply the technique to a scheme of Okamoto and Takashima [24, Section 4] in the ePrint version of this work [25]). This technique is not generic, our construction modifies the internal working of the underlying IPE scheme. The intuition is that to preserve correctness internal randomness needs to be stripped and replaced by a global one. This prevents an attacker from mixing and matching ciphertext components (or token components). Such mixing and matching is allowed for multi-input inner-product encryption. We compare the two notions in Section 3.1.

We implemented two versions of proximity search building on this form of $IPE_{fh,sk}$. The first is a direct application of the RProjC [16] scheme and the second is our new multi random projection version, called Multi Random Projection with Check or MRProjC. To benchmark, we encrypted a single reading of each individual ($\ell = 356$) from the ND0405 dataset [26, 27] which is a superset of the NIST Iris Evaluation Challenge [28]. Queries are drawn from other readings in the ND0405 dataset. This performance is summarized in Table 1 with search taking approximately 4 minutes. Our multi random projection technique reduces time for Setup by four orders of magnitude with minimal impact on the timings of the rest of the algorithms. This multi random projection technique makes proximity searchable encryption on a 350 biometric dataset feasible.

Distance Hiding Proximity Search. By design, for any searched value y, proximity search from $IPE_{fh,sk}$ allows the server to compute the distance [16] between y and all stored records.[3] This establishes a geometry on the space of stored records. If the server has side information on the stored records x_i, they may be able to reconstruct global geometry from the local geometry revealed by pairwise distances [31, 32]. While we are not aware of any leakage abuse attacks directly against proximity search, there are attacks against k-nearest neighbor databases [33, 34].[4] Distance allows one to easily compute the k-nearest points so attacks that can exploit this leakage apply. Like most leakage abuse attacks, the efficacy of these attacks depends on what the adversary knows about the stored data. We discuss these attacks more in Section 3.

For applications where such leakage is unacceptable (or the adversary has side information on the encrypted data), we show a transform from a predicate version of inner product encryption to proximity search that does not reveal

---

[3]Some prior work allows computation of approximate distance [29] using locality sensitive hashes [30], allowing the server to see how many hashes match, the number of matches approximates distance.

[4]Here we focus on attacks that apply to proximity searchable encryption. There is a rich history of leakage abuse attacks against different types of searchable encryption [35, 36, 37, 38, 39, 40, 33, 41, 34, 42, 43].

Scheme Name	IPE	IPE Type			MRProj used	Hide D	Operation Time			
		fh	sk	pred			Setup	BIndex	Trpdr	Search
RProjC	[16]	X	X				2M	10.8	.07	235 268
MRProjC	[16]	X	X		X		10.8	.08		241 268
MRProj (asym.)	[44]	X	X	X	X	X	10.8	22.4	12600	225
MRProj (sym.)	[44]	X	X	X	X	X	4.3	52	3580	

Table 1: Time (in seconds) for operations with ' = 356 records stored at n = 1024. All algorithms are naturally parallelizable. Timing for the single base scheme RProjC is interpolated from smaller vector lengths. BIndex encrypts the dataset at initialization time, Trpdr generates a search token, and Search nds the resulting indices. fh, sk and pred indicate that the underlying IPE scheme is respectively a function-hiding, secret key and/or predicate only scheme. Hide D indicates that the scheme does not reveal the distance between the stored value and the query. The symmetric version of MRProj uses the SS512 curve and the asymmetric version uses the MNT159 curve.

pairwise distance. A predicate IPE scheme produces ciphertexts ct_x and tokens tk_y which allow one to eectively check if $hx; yi = 0$ (instead of revealing the inner product). Barbosa et al. [44] recently proposed such a scheme that is a modication of Kim et al.'s construction [16]. Their construction simply removes the group elements that allow one to check the inner product, so we call this Random Projection or RProj. We call such a scheme an $IPE_{fh;sk;pred}$ scheme. Applying our multi random projection technique to this scheme yields a new predicate IPE scheme that we call Multi Random Projection or MRProj. $IPE_{fh;sk;pred}$ allows one to test if the inner product is equal to some value i as follows: add an $n + 1^{th}$ element as 1 to x, denoted x^0, and create $y_i = yjji$. Then, $hx^0; y_ii = hx jj -1; y jj ii = 0$ if and only if $hx; yi = i$. One can check all values in a set I by generating a token tk_y for each $i \in I$. Setting $I = fn 2 0; ::::; n 2 tg$, yields a proximity check (these tokens are permuted before being sent to the server). We show that with such a construction, an adversary cannot distinguish between two sets of encrypted records and queries as long as they have the same leakage discussed below.

The simplicity and generality of this construction is an advantage, it immediately benets from eciency improvements in inner product encryption and can be built from multiple computational assumptions.[5] However, the size of tk_y and the search time now grow linearly with t. For the iris t is usually around :3n.

Since the server can see if the same tk_y matches dierent records, when two records are both within distance t, the server learns if they match the same distance (but not the specic distance). Thus, the resulting proximity scheme leaks two pieces of information:

Access Pattern [35, 36] The set of records returned by the query. If x_i and x_j are both returned by a query it must be the case that $D(x_i; x_j)$ 2t.

---

[5]Throughout this work, we refer to proximity searchable encryption constructions by the name of the underlying IPE scheme. As an example, MRProj will be used to denote both Barbosa et al's IPE scheme with our multi random projection technique applied to it and the distance-hiding proximity search built using it.

Preventing attacks that only require access pattern usually requires oblivious RAM [45]. This is the high level approach taken by Boldyreva and Tang [46] in parallel work. They proposed a scheme that hides all leakage using oblivious data structures in conjunction with locality-sensitive hashes [30]. Their scheme is interactive, requiring several rounds of communication, but only uses symmetric key cryptography and is faster in a network with short round trip times.

Distance Equality Leakage For a database $x_1; \ldots; x_\ell$ and for a searched value y, if there are multiple records $x_i; x_j$ such that $D(x_i; y) \leq t$ and $D(x_j; y) \leq t$ then our scheme additionally reveals if $D(x_i; y) = D(x_j; y)$.

No information is leaked about data that is not returned (beyond that it was not returned). Biometrics are well spread[6], so one does not expect readings of two biometrics to be close to a query. As mentioned, the vector size has a large impact on the number of improper records that will be returned by a query (recall for $n = 64$, 40 improper records are returned, when $n = 1024$, :06 improper records are returned). Since MRProj only leaks when multiple records are returned it is critical to ensure an accurate system, underscoring the importance of our multi random projection approach enabling Setup for large n where high correctness is possible.

In RProjC and MRProjC, the server learns the pairwise distance between the query y and all records x_i. So in that setting, n only affects correctness, not security.

The search complexity of MRProj is roughly a multiplicative of $t \approx :3n$ slower than for MRProjC. See the difference in concrete timing in Table 1. For $n = 1024$, corresponding to a $t \approx 307$, the measured multiplicative overhead is only 52:5. Closing this performance gap is the main open problem resulting from this work; MRProj search is not fast enough.

Our analysis of MRProj is secure and correct with either a symmetric or asymmetric pairing. We implement both options. In our search implementation, the symmetric pairing instantiation is roughly 3 times faster than the asymmetric pairing instantiation. To the best of our knowledge, this is the first time that a function hiding inner product encryption has been analyzed in a symmetric pairing group, this analysis may be of independent interest. In Section 8 we posit additional avenues for improving search efficiency.

1.2. Differences between [1] and this work

A conference version of this work appeared at AsiaCCS 2022 [1]. There are two major differences between this manuscript and that article:

1. This work now considers symmetric bilinear groups throughout. All prior works including the conference version of this work [1] considered an asymmetric bilinear group. The evaluation section is also updated to contain

---

[6]Section 4 goes over the characteristics of the iris biometric in more details.

new timing results. The scheme based on symmetric bilinear groups improves search time by a factor of roughly 3.

2. We discuss in more details the multi random projection technique, and specially why it is not black-box applicable.

1.3. Organization

The rest of this work is organized as follows: Section 2 describes mathematical and cryptographic preliminaries, Section 3 reviews further related work, Section 4 describes the n vs accuracy tradeo for the iris and its impact on security, Section 5 introduces the multi random projection technique, Section 6 shows that $IPE_{fh,sk,pred}$ suces to build proximity search, Section 7 discusses our implementation, and Section 8 concludes.

2. Preliminaries

Let be the security parameter throughout the paper. We use $poly()$ and $negl()$ to denote unspecied functions that are polynomial and negligible in , respectively. For some $n \in N$, $[n]$ denotes the set $\{1, \ldots, n\}$. Let $x \overset{\$}{\leftarrow} S$ denote sampling x uniformly at random from the nite set S. Let $q = q() \in N$ be a prime, then G_q denotes a cyclic group of order q. Let x denote a vector over Z_q such that $x = (x_1, \ldots, x_n) \in Z_q^n$, the dimension of vectors should be apparent from context. Consider vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$, their inner-product is denoted by $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$. Let X be a matrix, then X^T denotes its transpose.

Hamming distance is dened as the distance between the bit vectors x and y of length n: $D(x, y) = |\{i \mid x_i = y_i\}|$. We note that if a vector over $\{0, 1\}$ is encoded as $x_{1,i} = 1$ if $x_i = 1$ and $x_{1,i} = -1$ if $x_i = 0$ then it is true that $\langle x_1, y_1 \rangle = n - 2D(x, y)$.

Our proofs rely on the Schwartz-Zippel lemma [47, 48]. We use the version from Kim et al.'s work [17, Lemma 2.9]:

Lemma 1 (Schwartz-Zippel Lemma). Fix a prime p and let $f \in F_p[x_1, \ldots, x_n]$ be an n-variate polynomial with degree at most d and which is not identically zero. Then,

$$\Pr[x_1, \ldots, x_n \overset{\$}{\leftarrow} F_p : f(x_1, \ldots, x_n) = 0] \leq d/p$$

We dene symmetric bilinear groups.

Denition 1 (Symmetric Bilinear Group). Suppose G_1 and G_T are two groups (respectively) of prime order q with generators $g_1 \in G_1$ and $g_T \in G_T$ respectively. We denote a value x encoded in G_1 with either g_1^x or $[x]_1$, we denote values encoded in G_T similarly. Let $e : G_1 \times G_1 \to G_T$ be a non-degenerate (i.e.: $e(g_1, g_1) \neq 1$) bilinear pairing operation such that for all $x, y \in$

8

$\mathbb{Z}_q$, $e([x]_1; [y]_1) = e(g_1; g_1)^{xy}$. Assume the group operations in G_1 and G_T and the pairing operation e are eciently computable, then $(G_1; G_T; q; e)$ denes an symmetric bilinear group.

As we show in Section 5, our scheme is secure in a symmetric bilinear generic group. However, we also present timing results with an asymmetric group that was used to argue the security of previous function-hiding inner product encryption schemes [16, 17, 44]. Proofs of security in a symmetric bilinear group extend to an asymmetric bilinear group. Correctness of our scheme follows in either setting.

2.1. Generic Group Model

The constructions presented in Figures 5 is based on a original construction proved secure in the asymmetric generic bilinear group model [49, 50]. However, we show security directly in the symmetric generic bilinear group model which is presented below. The particularity of the generic group model is to replace actual group elements by handles. Using these handles, the adversary is able to perform group and pairing operations. We adapt Kim et al.'s generic bilinear group oracle denition to the symmetric setting:

Denition 2 (Generic Bilinear Group Oracle). The generic bilinear group oracle is a stateful oracle dened as follows:

- Setup(1): Generate two fresh nonces $pp; sp \xleftarrow{\$} \{0,1\}$ and a prime q, and store them. Initialize an empty table Tab $= \{\}$ and set the internal state to subsequent call of Setup to fail. Finally, return $(pp; sp; q)$.

- Encode($k; x; i$): Receive $k \in \{0,1\}; x \in \mathbb{Z}_q$ and $i \in \{1; T\}$. If $k = sp$ return $\perp$. Else, generate a fresh nonce $h \xleftarrow{\$} \{0,1\}$ and add the entry $h \mapsto (x; 1)$ to table Tab. Return the handle h.

- Add($k; h_1; h_2$): Receive $k; h_1; h_2 \in \{0,1\}$.
    1. If $k = pp$ or one of the handles $h_1; h_2$ is absent from table Tab or $h_1; h_2$ do not map to values $(x_1; i_1)$ and $(x_1; i_2)$ with $i_1 = i_2$, return $\perp$.
    2. Compute $x = x_1 + x_2 \in \mathbb{Z}_q$.
        (a) If $h \mapsto (x; i_1)$ in Tab return h.
        (b) Else, generate a fresh handle $h \xleftarrow{\$} \{0,1\}$, set $h \mapsto (x; i_1)$ in Tab and return h

- Pair($k; h_1; h_2$): Receive $k; h_1; h_2 \in \{0,1\}$.
    1. If $k = pp$ or one of the handles $h_1; h_2$ is absent from table Tab or do not map to values $(x_1; 1)$ and $(x_2; 1)$ respectively, return $\perp$.
    2. Else, compute $x = x_1 x_2 \in \mathbb{Z}_q$
        (a) If $h \mapsto (x; T)$ in T return h.

9

(b) Else, generate a fresh handle $h \xleftarrow{\$} \{0,1\}$, set $h \mapsto (x, T)$ in Tab and return h

- ZeroTest(k, x): Receive $k, x \in \{0,1\}$. If $k = pp$ or h is absent from table T, return $?$. Else, return "zero" if $x \in \mathbb{Z}_q$ is 0 and "non-zero" otherwise.

As in previous works [16, 17, 44], we will analyze security by viewing each query as forming a formal polynomial. We re-state Remark 2.8 from Kim et al. [17]:

The generic bilinear group oracle is formally defined in terms of handles, however one can view oracle queries as formal polynomials. Each Encode query species a new formal variable for the polynomial. The adversary can then build terms for the polynomial by making Add and Pair oracle queries. The ZeroTest query outputs zero when the previously built polynomial evaluates to zero.

2.2. Inner Product Encryption

Functional encryption allows to compute a function on an encrypted input (the attribute), and retrieve the result without revealing more on the input. Predicate encryption is restricted to predicates, functions that output a single bit. While some works try to build schemes for general functionalities, others focus on specific ones. In the latter, one line of work aims to build efficient and secure schemes for inner products. In such schemes, the ciphertext and token encode vectors x and y, respectively, allowing to compute $\langle x, y \rangle$, the inner product between x and y, when running the decryption algorithm on the corresponding ciphertext and token. These schemes are called inner-product functional encryption schemes.

The predicate version of inner product encryption [51, 52] works in a similar manner, but instead of the decryption outputting the inner product value, it outputs 1 if $\langle x, y \rangle = 0$, 0 otherwise. [7] Secret-key predicate encryption with function privacy supporting inner products queries was first proposed by Shen et al. [52]. The scheme they presented is both attribute and function hiding, meaning that an adversary running the decryption algorithm gains no knowledge on either the attribute or the predicate.

We define secret-key inner production functional encryption ($IPE_{fh,sk}$) and secret-key inner product predicate encryption ($IPE_{fh,sk,pred}$) over the message space $\mathbb{Z}_q^n$.

Definition 3 (Secret key inner product functional encryption). Let $\lambda \in \mathbb{N}$ be the security parameter. Define $IPE_{fh,sk} = (\text{Setup}, \text{Encrypt}, \text{TokGen}, \text{Decrypt})$, a secret-key inner product functional encryption scheme over $\mathbb{Z}_q^n$ as

- $(pp, sk) \leftarrow \text{Setup}(1^\lambda)$: Generate public parameter pp and secret key sk.

---

[7] In some works, predicate inner product encryption outputs a message m when $\langle x, y \rangle = 0$ instead of a single bit.

- $ct_x \leftarrow$ Encrypt$(sk; x)$: Take secret key sk and input vector $x \in \mathbb{Z}_q^n$ and generate ciphertext ct_x.

- $tk_y \leftarrow$ TokGen$(sk; y)$: Take secret key sk and input vector $y \in \mathbb{Z}_q^n$ and generate token tk_y.

- $z \leftarrow$ Decrypt$(pp; tk_y; ct_x)$: Take public parameters pp, ciphertext ct_x and token tk_y and outputs value $z \in \mathbb{Z}_q$.

Correctness:. For any $x \in \mathbb{Z}_q^n; y \in \mathbb{Z}_q^n$,

$$\Pr\left[z = \langle x; y \rangle \;\middle|\; \begin{array}{l} ct_x \leftarrow \text{Encrypt}(sk;x) \\ tk_y \leftarrow \text{TokGen}(sk;y) \\ z \leftarrow \text{Decrypt}(pp;tk_y;ct_x) \end{array} \right] \geq 1 - \text{negl}():$$

Security of admissible queries:. Let $r = \text{poly}()$ and $s = \text{poly}()$. Any PPT adversary A has only negl() advantage in the $\text{Exp}^{\text{IPE}}_{\text{IND}}$ game (defined in Figure 1). Token and encryption queries must meet the following admissibility requirements, $\forall i \in [1; r]; \forall j \in [1; s]$,

$$\langle x_i^{(0)}; y_j^{(0)} \rangle = \langle x_i^{(1)}; y_j^{(1)} \rangle:$$

Definition 4 (Secret key inner product predicate encryption). Secret key in-ner product predicate encryption ($\text{IPE}_{\text{fh;sk;pred}}$) is defined similarly to secret key inner product functional encryption, with the difference that the output of the decryption algorithm is now $z \in \{0; 1\}$.

Correctness. For any $x \in \mathbb{Z}_q^n; y \in \mathbb{Z}_q^n$,

$$\Pr\left[z = \langle x; y \rangle \overset{?}{=} 0 \;\middle|\; \begin{array}{l} ct_x \leftarrow \text{Encrypt}(sk;x) \\ tk_y \leftarrow \text{TokGen}(sk;y) \\ b \leftarrow \text{Decrypt}(pp;tk_y;ct_x) \end{array} \right] \geq 1 - \text{negl}():$$

Security of admissible queries:. Let $r = \text{poly}()$ and $s = \text{poly}()$. Any PPT adversary A has only negl() advantage in the $\text{Exp}^{\text{IPE}}_{\text{IND}}$ game (defined in Figure 1). Token and encryption queries must meet one of the two following ad-missibility requirements, $\forall i \in [1; r]; \forall j \in [1; s]$,

$$\langle x_i^{(0)}; y_j^{(0)} \rangle = 0 \;\wedge\; \langle x_i^{(1)}; y_j^{(1)} \rangle = 0$$

or

$$\langle x_i^{(0)}; y_j^{(0)} \rangle = 0 \;\wedge\; \langle x_i^{(1)}; y_j^{(1)} \rangle = 0:$$

The above definition is called full security in the language of Shen, Shi, and Waters [52]. Note that this definition is selective (not adaptive), as the adversary species two sets of attributes and token values apriori. When discussing privacy it can be interesting to use a simulation-based security definition as it allows to specify exactly which amount of information is leaked.

We dene the following game between challenger C and adversary A:

1. C draws $\overset{\$}{} \in \{0;1\}$.
2. C computes $(sk;pp) \leftarrow$ Setup(1), sends pp to A.
3. For $1 \le i \le r$, A chooses attribute vectors $x_i^{(0)}; x_i^{(1)} \in Z_q^n$. 4.
   For $1 \le j \le s$, A chooses vectors $y_j^{(0)}; y_j^{(1)} \in Z_q^n$.
5. Denote

$$R := \left(x_1^{(0)}; x_1^{(1)}\right); \; ; \left(x_r^{(0)}; x_r^{(1)}\right); S$$
$$:= \left(y_1^{(0)}; y_1^{(1)}\right); \; ; \left(y_s^{(0)}; y_s^{(1)}\right):$$

6. A sends R and S to C.
7. A loses the game if R and S are not admissible.
8. A receives from C a list of ciphertexts

$$C^{()} := \left(ct_i^{()} \leftarrow \text{Encrypt}\left(sk; x_i^{()}\right)\right)_{i=1}^r$$

   and a list of tokens

$$T^{()} := \left(tk_j^{()} \leftarrow \text{TokGen}\left(sk; y_j^{()}\right)\right)_{j=1}^s$$

9. A returns $^0 \in \{0;1\}$.
10. A's advantage is

$$\overline{Adv_A^{Exp_{IND}^{IPE}}()} = \left| Pr[A(1; T^{(0)}; C^{(0)}) = 1] \quad Pr[A(1; T^{(1)}; C^{(1)}) = 1] \right|$$

Figure 1: Denition of Exp_{IND}^{IPE} for inner product encryption.

Denition 5 (Simulation-based security for IPE). Let IPE = (Setup; TokGen, Encrypt; Decrypt) be an inner product encryption scheme over Z_q^n. Then IPE is SIM-secure if for all P P T adversaries A, there exists a simulator S such that for the experiment Exp_{SIM}^{IPE} described in Figure 2, the advantage of A $(adv_A^{Exp_{SIM}^{IPE}})$ is $\left| Pr[1 \quad Real_{IPE;A}(1)] \quad Pr[1 \quad Ideal_{IPE;A}(1)] \right| \quad negl():$

Kim et. al. [17, Remark 2.5] show that Denition 5 implies Denition 3.

2.3. Proximity searchable encryption

In this section we dene proximity searchable encryption (PSE), a variant of searchable encryption that supports proximity queries.

Denition 6 (History). Let $X \in M$ be a list of keywords drawn from space M, let F be a class of predicates over M. An m-query history over W is a tuple History $= (X; F)$, with $F = (f_1; \; ; f_m)$ a list of m predicates, $f_i \in F$.

$\text{Real}_{\text{IPE};A}(1)$
$(sk; pp) \quad \text{IPE:Setup}(1)$
$b \quad A^{\text{IPE:TokGen}(sk;);\text{IPE:Encrypt}(sk;)}(1)$
Output b

$\text{Ideal}_{\text{IPE};A}(1)$
$(sk; pp) \quad \text{IPE:Setup}(1) \; b$
$\quad A^{S(())}(1) \quad$ Output b

Figure 2: Denition of experiment $\text{Exp}_{\text{SIM}}^{\text{IPE}}$. denotes the information leakage received by the simulator S such that $(i; j) = hx_i; y_j i = 0$ for all $i; j$.

Denition 7 (Access pattern). Let $X \in M$ be a list of keywords. The access pattern induced by an m-query history History $= (X; F)$ is the tuple $\text{AccPatt}(\text{History}) = (f_1(X); ; f_m(X))$.

Denition 8 (Distance Equality). Let $\text{History}^{(0)}; \text{History}^{(1)}$ be m-query histories for predicates of the type $f_{y;t}(x) = (D(x; y) \overset{?}{} t)$.

Let $\text{DisEq}(\text{History}^{(0)}; \text{History}^{(1)}) = 1$ if and only if for each j it is true that

$$
(i; k)_{(D(x_i^{(0)};y_j^{(0)})=D(x_k^{(0)};y_j^{(0)})\wedge D(x_i^{(1)};y_j^{(1)})=D(x_k^{(1)};y_j^{(1)}))}^{(D(x_i^{(0)};y_j^{(0)})=D(x_k^{(0)};y_j^{(0)})\wedge D(x_i^{(1)};y_j^{(1)})=D(x_k^{(1)};y_j^{(1)}))} \quad ;
$$

is the empty set.

Denition 9 (Proximity Searchable Encryption). Let

- $2 N$ be the security parameter,
- $DB = (M_1; ; M\cdot)$ be a database of size '',
- Keywords $X = (x_1; ; x\cdot)$, such that $x_i \in Z_q$ relates to M_i,
- $F = ff_{y;t} \; j \; y \in Z_q^n; \; t \in Ng$ be a family of predicates such that, for a keyword $x \in Z_q^n$, $f_{y;t}(x) = 1$ if $D(x; y) \; t$, 0 otherwise.

The algorithms $\text{PSE} = (\text{PSE:Setup}; \text{PSE:BIndex}, \text{PSE:Trpdr}; \text{PSE:Search})$ denes a proximity searchable encryption scheme:

- $\text{PSE:Setup}(1) \; ! \; (sk; pp)$, •
  $\text{PSE:BIndex}(sk; X) \; ! \; I_X$,
- $\text{PSE:Trpdr}(sk; f_{y;t}) \; ! \; tk_{y;t}$, and
- $\text{PSE:Search}(pp; Q_{y;t}; I_X) \; ! \; J_{X;y;t}$.

We require the scheme to have the following properties:

We dene the following game between challenger C and adversary A:

1. C draws $\overset{\$}{} \, \in \{0;1\}$.
2. C computes $(sk; pp) \leftarrow PSE:Setup(1)$ and sends pp to A.
3. A chooses and sends $History^{(0)}; History^{(1)}$ to C.
4. A loses the game if

   $AccPatt(History^{(0)}) = AccPatt(History^{(1)}) \wedge DisEq(History^{(0)}; History^{(1)}) = 0$

5. A receives $I^{()}$ and $Q^{()}$ from C. 6.
A outputs $^{0} \in \{0;1\}$.
7. A's advantage in the game is:

   $$Adv_{A}^{Exp_{IND}^{PSE}}() = \left| Pr[A(1; I^{(0)}; Q^{(0)}) = 1] \quad Pr[A(1; I^{(1)}; Q^{(1)}) = 1] \right|$$

Figure 3: Denition of Exp_{IND}^{PSE}.

Correctness. Dene $J_{X;y;t} = \{i \mid f_{y;t}(x_i) = 1; x_i \in X\}$. PSE is correct if for all X and $f_{y;t} \in F$:

$$Pr\left[J^{C} = J_{X;y;t} \, \middle| \, \begin{array}{l} I_{X} \leftarrow PSE:BIndex(sk;X) \\ Q_{y;t} \leftarrow PSE:Trpdr(sk;f_{y;t}) \\ J \leftarrow PSE:Search(pp;Q_{y;t};I_{X}) \end{array} \right] \geq 1 \quad negl():$$

Security for Admissible Queries. Any PPT adversary A has only negl() advantage in the experiment Exp_{IND}^{PSE} dened in Figure 3, for $' = poly()$ and $m = poly()$.

3. Further Related Work

3.1. Functional encryption

We review further related work on functional and predicate encryption.

Function privacy for public key schemes. Building distance-hiding proximity searchable encryption from inner product encryption requires the latter to be function-hiding. The PSE scheme presented in this work is secret key, but one could want to build a public-key variant. Such a scheme would require public key function-hiding IPE.

Achieving function privacy for public key functional encryption is not straight-forward. The adversary can encrypt ciphertexts on its own and run the function on the corresponding inputs, allowing them to learn information on the function's behavior.

Boneh et al. [53] presented a function-hiding identity-based encryption scheme which requires token inputs to be sampled from a distribution with super-logarithmic min-entropy. Their function privacy notion then requires that for

y sampled from such a distribution, the corresponding token tk_y must be indistinguishable from a token tk_u where u was independently and uniformly sampled. In a subsequent work [54], Boneh et al. consider the notion of function-hiding public key subspace-membership encryption which supports subspace-membership predicates, a generalization of inner products.

Multi-input functional encryption. Readers familiar with multi-input functional encryption for the inner product may notice some similarities between this line of work and the multi-random projection technique.

Multi-input functional encryption (MIFE) [55] is a generalization of functional encryption that supports functions with multiple inputs. Token generation works as in standard functional encryption but encryption allows for multiple users to encrypt their inputs independently. Decryption then takes the token tk_f and the multiple ciphertexts $ct_1, \ldots, ct_n$ and computes $f(x_1, \ldots, x_n)$. In the case of multi-input IPE [56, 57], the supported function is of the form

$$\sum_{i=0}^{n} \langle x_i, y_i \rangle.$$

The main challenge to building MIFE schemes is to combine ciphertexts generated using independent randomness in a secure manner.

Although multi-input IPE and multi-random projection seem to achieve similar goals they are dierent concepts. In multi-input IPE, one should be able to decrypt ct_1 with any available value for $ct_2, \ldots, ct_n$. In the multi-random projection all components of a single ciphertext should only work with each other. In other words, multi-input IPE allows for mix-and-matches whereas MRProj must prevent it. Mix-an-matching would indeed allow the adversary to create ciphertexts and tokens that are not admissible, resulting in a break of security. To prevent this, multi-random projection requires global randomness to tie the multiple ciphertexts (respectively tokens) together.

3.2. Proximity search

We review further related work on proximity search. Li et al. [58], Wang et al. [59] and Boldyreva and Chenette [60] reduced proximity search to keyword equality search. These works propose two complimentary approaches:

1. When adding a record x_i to a database, also insert all close values as keywords, that is $\{x_j \mid D(x_i, x_j) \leq t\}$ are added as keywords associated to x_i.
2. The second approach requires searchable encryption supporting disjunctive search. Disjunctive search generally allows to perform search using a set of keywords, returning a record when at least one of those keywords is a match. This approach inserts just x_i, but when searching for y it searches for the disjunction $\bigvee_{x_i \mid D(x_i, y) \leq t} x_i$.

Either approach can be instantiated using a searchable encryption scheme that supports disjunction over keyword equality (inheriting any leakage). However,

for biometrics, the number of keywords $\bigcup_\chi |D(\chi; y)| t \{x_i\}$ usually grows exponentially in t. In existing disjunctive schemes, the size of the query grows with the size of the disjunction [15], making this approach only viable for constant values of t.

Kuzu et al.'s [29] solution relies on locality sensitive hashes [30]. A locality sensitive hash ensures that close values have a higher probability to produce collisions than values that are far apart. Thus, proximity search can be built from any scheme supporting disjunctive keyword equality, inheriting any leakage. The server learns the number of matching locality sensitive hashes for each record (which is expected to be more than 0). The number of matching locality sensitive hashes is a proxy for the distance between the query value and the records. More matching locality sensitive hashes implies smaller distance. This allows the server to establish the approximate distance between each record and the query.

Zhou and Ren [61] propose a variant of inner product encryption that reveals if the distance is less than t only. However, their security is based on Ax_i and yB hiding x_i and y for secret square matrices A and B. Security is heuristic with no underlying assumption or proof of information theoretic security.

Abuse Attacks. Searchable encryption achieves acceptable performance by leaking information to the server. See Kamara, Moataz, and Ohrimenko for an overview of leakage types in structured encryption [62]. The key to attacks is combining leakage with auxiliary data, such as the frequency of values stored in the data set. Together these sources can prove catastrophic { allowing the attacker to recover either the queries being made or the data stored in the database. We consider attacks that rely on injecting les or queries [63] to be out of scope. Common, attackable, relevant leakage proles are:

1. Response length leakage [37, 40] Often known as volumetric leakage, the attacker is given access to only the number of records returned for each query. Based on this information, attacks cross-correlate with auxiliary information about the dataset, and identify high frequency items in both the encrypted database and the auxiliary dataset.
2. Query equality leakage [38] the attacker is able to glean which queries are querying the same value, but not necessarily the value itself. Attacks on this prole rely on having information about the query distribution, and much like the response length leakage attacks, match that with auxiliary information based on frequency.
3. Access pattern leakage [35, 36] here the attacker is given knowledge if the same dataset element is returned for dierent queries. This allows the attacker to build a co-occurrence matrix, mapping which records are returned for pairs of queries. Based on the frequencies of the co-occurrence matrix for the encrypted dataset, and the co-occurrence matrix for the auxiliary dataset, the attack can identify records.

Recent attacks have targeted the geometry present in range search [39, 64, 40, 42, 43]. Building on the co-occurrence matrix (available with access pattern

leakage) consider the case when records $a; b; c$ are returned by a rst query and $c; d; e$ are returned by a second query. One can immediately infer that the comparison relation between a and d is the same as the comparison relation between b and e. As more constraints of this type are collected one can build an ordering of all records (up to reection).

In two (or three) dimensional Euclidean space, trilateration has been practiced for hundreds of years: one is assumed to know the location of $x_1; :::; x_k$ and the pairwise distances $D(x_i; y)$ and is trying to nd the location of y. Determining the location of y requires k to be one larger than the dimension. The problem is more dicult but well studied for approximate distances [65]. Simi-lar ideas can be applied in discrete metrics with each learned distance reducing the set of possible y. In the Hamming metric of dimension n, k = (n) suces [66, 67, 68].

4. Iris Statistics and Leakage

This section introduces iris feature extractors and shows that reducing the length of the feature extractor harms the uniqueness of the resulting biometric. Reduced uniqueness harms both the correctness (because the wrong set of irises is returned) and security of the MRProj construction (because the server learns information about returned irises). Daugman [18, 19] introduced the seminal iris processing pipeline. This pipeline assumes a near infrared camera. Iris images in near infrared are believed to be independent from the visible light pattern; the near-infrared iris pattern is epigenetic, irises of identical twins are believed to be independent [19, 69]. Traditional iris recognition consists of three phases:

Segmentation takes the image and identies which pixels should be included as part of the iris. This produces a $\{0; 1\}$ matrix of the same size as the input image with 1s corresponding to iris pixels.

Normalization takes the variable size set of iris pixels and maps them to a xed size rectangular array. This can roughly be thought of as unrolling the iris.

Feature Extraction transforms the rectangular array into a xed number of features. In Daugman's original work this consisted of convolving small areas of the rectangle with a 2D wavelet. Modern feature extractors are usually convolutional neural networks.

In identication systems the tradeo is between FRR and FAR. FRR is how frequently readings of the same biometric are regarded as dierent. FAR is how frequently readings of dierent biometrics are regarded as the same. As described above, when one wishes to match a biometric y against a database one considers matches as the set $\{x_i | D(x_i; y) t\}$ for some metric D and distance parameter t. Selecting a small t increases FRR and reduces FAR. Before investigating the dependence on feature vector length and the FRR/FAR tradeo we introduce the feature extractor and dataset used in this analysis.

17

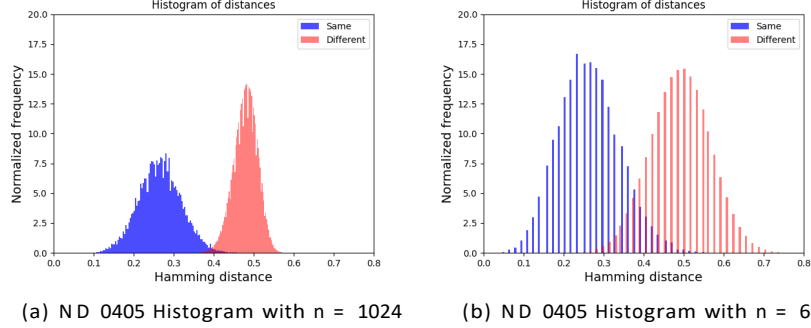(a) N D 0405 Histogram with n = 1024   (b) N D 0405 Histogram with n = 64

Figure 4: Hamming distance distribution for images from the same iris in blue, and dierent irises in red. Histograms are produced using ThirdEye [21]. Resulting histograms for the N D 0405 dataset. Figure 4(a) shows the histogram when n = 1024 with a small overlap between distances comparisons of the same iris and dierent irises. This overlaps is increased substantially when n = 64 in in Figure 3b). Figure 3b) is produced using the E method.

Feature Extractor. For the feature extractor, we use the recent pipeline called ThirdEye [70, 21], which is publicly available [71]. The software produces a 1024 dimensional real valued feature vector. We convert this to a binary vector by setting $f_i^0 = 1$ if $f_i > \text{Exp}[f_i]$ where the $\text{Exp}[f_i]$ is the expectation of the individual feature, otherwise $f_i^0 = 0$. We train the feature extractor as specied in [21].

Biometric Database. There are many iris datasets collected across a variety of conditions. In this work we use the NotreDame 0405 dataset [26, 27] which is a superset of the N IST Iris Evaluation Challenge [28]. This dataset consists of images from 356 biometrics (we consider left and right eyes as separate biometrics) with 64964 images in total. (See Appendix 4.2 for similar results with the I I T D dataset [72].) Figure 4(a) shows the histograms for the testing portions of the feature extractor outputs. The blue histogram contains comparisons between dierent readings of the same biometric while the red histogram contains comparisons between dierent biometrics. Let $t^0 = t{=}1024$ be the fractional Hamming distance, the F R R is the fraction of the blue histogram to the right of t^0 and the FA R is the fraction of the red histogram to the left of t^0. There is overlap between the red and blue histogram indicating that there is a tradeo between F R R and FA R.

4.1. Performance of Biometric Identication with Small Dimension

   The eciency of I P E based proximity search critically depends on the number of features n (see Table 5). In our experiments we estimate Setup for n = 1024 for the schemes of K i m et al. [17] and Barbosa et al. [44] to take 23 days on a modern server machine (see details in Section 7). It is tempting to consider statistical methods to produce feature vectors of reduced size. We show this comes at a cost to the quality of the resulting feature vectors. This

18

motivates our approach to reduce the complexity of Setup in Section 5. Our analysis consists of two major parts:

1. We compare dierent mechanisms for reducing the size of feature vectors using n = 64 as the target dimension.
2. Using the best feature reduction mechanism we compare the FRR/FAR tradeo for n < 1024, showing direct impacts for the correctness and security of the resulting biometric search.

4.1.1. Dimensionality Reduction Method

We consider four dierent mechanisms for dimension reduction and consider their impact on FRR/FAR. For all techniques, the most important phenomena is that variance of Dierent comparisons increases as the sample size decreases.[8] Compare Figure 4(a) and Figure 4(b). This makes the tails of Same and Different wider, leading to worse identication. The four mechanisms we consider are[9]:

Random Sample Select a random subset of positions of size 64 and use this as the feature extractor. We denote this technique by R-64 (for random).

Error Rate Minimization Hollingsworth et al. [74] and Bolle et al. [75] propose the concept of \fragile bits" which are more likely to be susceptible to bit ips. Their work is based on the Gabor based feature extractor (described at the beginning of this section) while ThirdEye [21] is a convolutional neural network.

We select the 64 bits which have the least probability of ipping. Results for this approach are shown in Table 2 and denoted by S-64 (for stable).

Surprisingly, this approach is worse than random sampling. We believe this approach to be appropriate for the Gabor based feature extractor since it produces large number of noisy features due to noise in dierent readings of an iris. This is in contrast to our feature extractor which outputs a succinct feature vector where the CNN tries to make individuals features independent.

Error Delta Maximization This approach uses bits which maximize the difference between the means of the intra and inter class distributions. These are bits where the dierence between intra class and inter class error is the highest. That is, we select the bits that maximize the following dierence:

$$\max_{i} \Pr_{x,y \leftarrow Different}[x_i = y_i] \quad \Pr_{x,y \leftarrow Same}[x_i = y_i]$$

---

[8] This is consistent with previous observations that sampling from the iris red histogram behaves similarly to a binomial distribution where the number of trials is proportional the included entropy of the iris [73].

[9] For all experiments we computed the mechanism four times and report the average in Table 2.

					False Accept Rate (FAR)						
Size	0	.01	.02	.03	.04	.05	.06	.07	.08	.09	.10
1024	.50	.03	.02	.01	.01	.01	.01	.01	.01	0	0
R-64	.99	.38	.29	.24	.22	.18	.17	.16	.14	.13	.12
S-64	1	.61	.61	.51	.41	.41	.41	.32	.32	.32	.26
E-64	.97	.30	.24	.18	.14	.14	.10	.10	.10	.07	.07
T-64	.96	.27	.16	.13	.13	.09	.09	.06	.06	.06	.04

Table 2: False reject rate (FRR) for the ND0405 datasets, for output size n = 64 and for dierent dimensionality reduction techniques. Queries are drawn from Same distribution. We vary a threshold t and report the FRR when allowing for the corresponding $\overline{FAR}$. The original n = 1024 system is presented for comparison.

Here, \Same" indicates that the readings x, y come from the same iris and\Dierent" means that they came from distinct irises. The intuition is that bits are the most useful as they maximize the dierence in probability of error between the same and dierent comparisons. The hope is to overcome the weakness of the prior approach which did not consider the entropy of bits across dierent biometrics. The top 64 bits are used. This approach is denoted by E-64 (for error). This approach improves over both R and S techniques.

Training Network Lastly, we train the ThirdEye architecture [21] from scratch to output a smaller feature vector of size n = 64 for both datasets. Essentially we train a new feature extractor on the same training data to reduce dimensions. The feature extractor remains the same but is now constrained to learn 64 features. This is achieved by changing the number of neurons in the second last layer of our convolutional neural network. We can expect this to perform better than random sampling since the feature extractor is explicitly learning to classify using 64 features. We use T-64 (for train) to denote this technique.

Results are summarized in Table 2. The E and T techniques outperform the R and S techniques. Going forward we use the E dimensionality reduction technique for the rest of this work because it is simpler to compute for dierent vector sizes.

4.1.2. Impact of reducing n

We now show that decreasing n using the E method hurts the identication quality of the iris biometric. First we note that an FRR of :10 requires a distance tolerance of t :3n (see the histograms in Figure 4). However, comparisons between dierent irises are tightly centered around t = :5n. This means for a dataset $\{x_i\}_{i=1}'$ for most pairs x_i, x_j there exists some value x such that $D(x_i, x)$ t and $D(x_j, x)$ t. This means for most pairs x_i, x_j, there is some query that will cause them both to be returned.

The goal of this subsection is to understand behavior on actual queries. We consider a distribution over x of dierent readings of individuals stored in

	Vector Length							
ACount	64	96	128	256	384	512	768	1024
Avg.	40.8	34.5	13.0	6.03	3.86	1.03	.53	.06
2	.75	.74	.42	.23	.17	.083	.076	.019

Table 3: Eect of dimensionality reduction on the correctness and security of the resulting biometric search system. ACount is the average number of improperly records when searching for a biometric that is in the dataset. All feature extractors with n < 1024 use the E method to select features.

the dataset to see how frequently multiple records are returned. Recall that multiple records being returned impacts the system correctness for both the MRProjC and MRProj constructions. It additionally aects leakage for MRProj. For these analysis we consider the ND-0405 dataset with the E mechanism for reducing the size of a feature vector (see the previous subsection).

We use the value ACount, which represents how frequently a record of a dierent biometric would be returned by an in use search system, to evaluate the impact of the dimensionality reduction. To achieve both correctness and security, one needs ACount to be as close to 0 as possible, assuming randomly distributed queries.

We consider correctness of the system at dierent feature vector lengths n. We select a random reading of each biometric to represent the encrypted dataset. We rst select a t that yields at most 10% FRR (for comparisons of the same iris on the training dataset). We then use the following procedure:

1. Initialize matrix $C_{i;j} = 0^{356 \times 356}$.
2. Pick $I \subseteq \{1, \ldots, 356\}$ of size 150 randomly. 3.
   For each i in I:
   (a) Select 3 random readings of iris i, denoted x_i (removing reading that is encrypted):[10]
   (b) For all j if $D(x_i, x_j) \leq t$ and $D(x, x_i) \leq t$ $C_{i;j} = C_{i;j} + 1$.
4. Compute ACount = $\sum_{i=0}^{355} \sum_{j=0; j \neq i}^{355} C_{i;j} / (3 \cdot 150)$:

We ran this experiment 40 times and report the mean and standard deviation of ACount in Table 3. As one can see keeping a vector size of n = 1024 has a three order of magnitude reduction in the average number of improperly returned records, underscoring the importance of inner product encryption to work with large n.

Leakage on readings of the same iris. There are two types of biometric databases, those which associate a single reading x_i of a biometric with each record r_i and those where multiple readings of a biometric $x_{i;1}, \ldots, x_{i;k}$ are associated with a single record. Until now, we've implicitly assumed that the database has only

---

[10] Every iris in the ND0405 dataset has at least 4 readings so this is the maximum number of queries that will have an equal number of readings from the size 150 subset.

FRR	False Accept Rate										
Size	0	.01	.02	.03	.04	.05	.06	.07	.08	.09	.10
1024	.70	1	1	1	1	1	1	1	1	1	1
512	.57	1	1	1	1	1	1	1	1	1	1
256	.47	.99	1	1	1	1	1	1	1	1	1
192	.48	.99	1	1	1	1	1	1	1	1	1
128	.54	.99	.99	1	1	1	1	1	1	1	1
96	.40	.99	.99	.99	1	1	1	1	1	1	1
64	27	.97	.99	.99	.99	.99	.99	1	1	1	1

Table 4: TAR for dierent output sizes and probabilities of leakage for the IITD Dataset. Summary of FAR for queries drawn from Same distribution for noise tolerance parameters. We vary a threshold t, report the FRR when FAR is as listed. All sizes use the R methodology.

one reading of a biometric. We now briey consider the implications of leakage between readings of the same biometric. That is, $x_{i;1}; \ldots; x_{i;k}$ are readings from the same biometric and associated with a record r_i in the biometric database. First note that $x_{i;}$ and $x_{i;}$ are likely to be close together (because readings of the same biometric are similar).

One may able to infer information about $x_{i;1}; \ldots; x_{i;k}$ from access pattern and distance equality leakage. One may be able to learn the relative position-ing of the dierent readings by which values I are return by a query y (if it is not all values). Similarly, we expect the adversary to learn distance equality leakage for the entire set $x_{i;1}; \ldots; x_{i;k}$. Both of these leakage proles allow an adversary to construct geometry of a biometric's dierent readings. This may allow the adversary to determine the type of noise present in that individual's biometric. It may be possible to use noise rates to draw conclusions about sen-sitive attributes about the corresponding person. Biometric systems frequently demonstrate systemic bias [76]. As one example most datasets draw from vol-unteer undergraduates students. Systems accuracy varies based on sensitive attributes such as gender, race, and age (see [76, Table 1]). Thus one may be able to infer sensitive attributes based on the relative size of jIj=k.

If one stores multiple readings, it seems important to use cryptographic techniques to hide such leakage. A potential solution is to instead store a single reading that is the average of the multiple readings [77] and make other values associated data that are not searchable.

4.2. Statistical Analysis for IITD Dataset

The IITD dataset which consists of 224 persons and 2240 images. The IITD dataset is considered \easier" than the ND0405 dataset because images are collected in more controlled environments leading to less noise and variation between images. Table 4 shows the FAR/FRR tradeo for IITD dataset akin to Table 2. We additionally measured the number of improperly returned records as in Table 3; improper records where only observed for length 64. Since IITD is easier than ND0405, this indicates that the needed biometric dimension depends on collection conditions.

5. Multi Random Projection IPE

As described in the Introduction, we show a general technique improving Setup eciency for IPE schemes where ciphertexts and tokens are projected into dual vector spaces by a pair of matrices $A; A^{-1}$. When applied to a secret-key function hiding predicate IPE (respectively secret-key function hiding IPE), this technique yields an IPE scheme with the same security properties. We call this multi random projection technique. The key idea is to create multiple pairs of matrices of smaller dimension for subvectors of the inputs. These independent encodings are then combined with an additive secret sharing of 0 in the encryption so that computation with ciphertexts and tokens is only useful when using all of the components. Without this additional step, an adversary could discard some subvectors of the inputs and still learn the inner products of the remaining ones. In this section we show security of the technique when applied to the RProj scheme of Barbosa et al. [44, Section 4].[11]

This scheme from Barbosa et al. is built upon an asymmetric bilinear pairing. In the conference version of this work, we applied the multi-random projection technique to this asymmetric scheme. This work will shows the construction is secure with a <u>symmetric</u> bilinear pairing. This analysis allows one to choose between a Type 1, 2 or 3 pairing[12], whichever provides the best performance. In our experiments using the Charm library [78], presented in Section 7, the symmetric pairing is more ecient. Other recent implementations using Charm [79, 80, 81] found superior performance with a symmetric pairing.

Construction. The construction is in Figure 5. We rst argue correctness and then security. For security, we show the scheme satises a stronger simulation-based denition of security, as in the work of Barbosa et al. [44]. Unlike Kim et al. [16, 17] and Barbosa et al. [44] we work directly with symmetric bilinear groups. They both argued security assuming asymmetric bilinear groups.[13]

Correctness. First note that $\langle x; y \rangle = \sum_{\ell=1}^{P} \langle x_\ell; y_\ell \rangle$, and thus

$$\prod_{\ell=1}^{N} \prod_{i=1} e(tk_\ell[i]; ct_\ell[i]) = g_T^{\sum_{\ell=1}^{P} (x_\ell^0)^T B_\ell B_\ell'' (y_\ell)_\ell} = g_T^{\sum_{\ell=1}^{0}}$$

$$(x_\ell^0)^T \prod^P (y_\ell^0) = g^{\sum_{\ell=1}^{P} \ell + \langle x_\ell; y_\ell \rangle}$$

$$= g_T^{\langle x; y \rangle + \sum_{\ell=1}^{P} \ell} = g_T^{\langle x; y \rangle}$$

If $\langle x; y \rangle = 0$ then $\prod_{\ell=1}^{N} \prod_{i=1} e(tk_\ell[i]; ct_\ell[i]) = e(g_1; g_2)^0 = 1$; which is the iden-tity element in G_T and is easily detectible and $>$ Decrypt(pp; tk; ct) with

---

[11] Functional encryption for orthogonality (OFE) as dened by Barbosa et al. is equal to predicate inner product encryption, as dened in this work.

[12] Type 1 pairing denotes a symmetric bilinear pairing whereas type 2 and 3 are asymmetric bilinear pairings.

[13] Both symmetric and asymmetric pairings work for functionality. For security, a symmetric pairing suces.

23

Setup($1; n;$):

1. Sample a symmetric bilinear group $(G_1; G_T; q; e)$ and choose generator $g \in G_1$.
2. Output $pp = (G_1; G_T; q; e; n;)$ as public parameters and $sk = (g; \{B_\ell; B_\ell g\}_{\ell=1})$.

TokGen($pp; sk; y$):

1. Sample $\overset{\$}{\leftarrow} Z_q$.
2. Split input $y \in Z_q^n$ into subvectors y_ℓ of size $\lceil n=e \rceil$ and pad with zeroes if needed.
3. For $1 \leq \ell$, dene $y^0 = 1 \| y_\ell$ and set $tk_\ell = [(y^0)^T B_\ell]_1$. 4.
   Output $tk = (tk_1; ; tk)$.

Encrypt($pp; sk; x$):

1. Sample $\overset{\$}{\leftarrow} Z_q$.
2. Split input $x \in Z_q^n$ into subvectors x_ℓ of size $\lceil n=e \rceil$, and pad with zeroes if needed.
3. For $1 \leq \ell \leq 1$, sample $\ell \overset{\$}{\leftarrow} Z_q$ then set $= \overset{P}{\underset{\ell=1}{}} 1 \ell$.
4. For $1 \leq \ell$ dene $x^0 = \ell \| x_\ell$ and set $ct_\ell = [(x^0)^T B_\ell]_1$. 5. Output
   $ct = (ct_1; ; ct)$.

Decrypt($pp; tk; ct$):

1. Compute $z = \prod_{\ell=1}\prod_{i=1} e(tk_\ell[i]; ct_\ell[i])$.
2. Return $\top$ if z is equal to $1 \in G_T$, $\perp$ otherwise.

Figure 5: Construction of MRProj.

probability 1. If $\langle x; y \rangle = 0$, then the probability that $\top \leftarrow$ Decrypt($pp; tk; ct$) is $Pr[\langle x; y \rangle = 0] \leq 2=q$:

We argue that the scheme in Figure 5 satises Denition 5.

Theorem 1. In the Generic Group Model (Denition 2) for symmetric bilinear groups the construction in Figure 5 is a secure $IPE_{fh;sk;pred}$ scheme according to Denition 5.

Proof of Theorem 1. Our scheme builds on the scheme of Barbosa et al. [44] built in turn on the work Kim et al.[17]. Our proof uses similar denitions of formal variables. The scheme works by having a challenger interact with a simulator S and two oracles, O^0_{TokGen} and $O^0_{Encrypt}$ in the ideal scheme, and a pair of oracles, O_{TokGen} and $O_{Encrypt}$, in the real scheme. For this proof, we will build the simulator S which can correctly simulate the distribution of tokens and ciphertexts only using the predicate evaluation on whether the inner product of the two vectors is 0. This information is supplied to the simulator by the oracles

O^0_{TokGen} and O^0_{Encrypt} to match the functionality of the encryption scheme. This is the information leakage described in Figure 2.

Inner-product collection. Let i, j be shared counters between the token generation and encryption oracles. Let $x^{(i)} \in Z_q^n$ and $y^{(j)} \in Z_q^n$ denote respectively the adversary's i^{th} query to the token generation oracle and j^{th} query to the encryption oracle. The collection of mappings C_{ip} is defined as

$$
C_{ip} = \begin{cases}
(i, j) \mapsto 0 & \text{if } \langle x^{(i)}, y^{(j)} \rangle = 0 \\
(i, j) \mapsto 1 & \text{otherwise.}
\end{cases}
$$

Formal variables. The simulator constructs formal variables for the unknowns of the system in order to respond as correctly as possible. Consider the following notation:

- Let Q be the maximum number of queries made by an adversary.

- Let and N be as in the construction in Figure 5.

- For all $i \in [Q]$, $\ell \in []$ and $k \in [N]$, let $\hat{\alpha}^{(i)}, \hat{\beta}^{(i)}, \hat{x}_{\ell;k}, \hat{y}^{(j)}_{\ell;k}$ represent the hidden variables $\alpha^{(i)}, \beta^{(i)}, x_{\ell;k}, y^{(j)}_{\ell;k}, $

- Let $\hat{b}_{\ell;k;m}$ and $\hat{b}_{\ell;k;m}$ represent the entry in position (k, m) of the B_ℓ and B_ℓ matrices respectively,

- Let $\hat{\gamma}^{(i)}_\ell$ be the formal variables for $\gamma^{(i)}_\ell$, where the simulator tracks the constraints that for each $i \in [Q]$; $\sum_{\ell = 1}^{} \hat{\gamma}^{(i)}_\ell = 0$, and

- Let $\hat{s}^{(i)}_{\ell;m}$ and $\hat{t}^{(i)}_{\ell;m}$ represent formal polynomials as constructed below,

$$
\hat{s}^{(i)}_{\ell;m} = \sum_{k=1}^{N} \hat{y}^{0(i)}_{\ell;k} \hat{b}_{\ell;k;m} = \hat{b}_{\ell;1;m} + \sum_{k=2}^{N} \hat{y}^{(i)}_{\ell;k} \hat{b}_{\ell;k;m} \tag{1}
$$

$$
\hat{t}^{(i)}_{\ell;m} = \sum_{k=2}^{N} \hat{x}^{0(i)}_{\ell;k} \hat{b}_{\ell;k;m} = \hat{x}^{(i)}_{\ell;1;m} \hat{b}_{\ell;k 1} + \sum \hat{x}^{(i)}_{\ell;k;m} \hat{b}_{k=1} \tag{2}
$$

Then the universe of formal variables is $U = R \cup T$, where

$$
R = \left\{ \hat{\alpha}^{(i)}; \hat{\beta}^{(i)} \right\}_{i \in [Q]} \cup \left\{ \hat{s}^{(i)}_{\ell;m}; \hat{t}^{(i)}_{\ell;m} \right\}_{i \in [Q]; \ell \in []; m \in [N]}
$$

and

$$
T = \left\{ \hat{\alpha}^{(i)}; \hat{\gamma}^{(i)}_\ell \right\}_{i \in [Q]}
$$
$$
\cup \left\{ \hat{x}^{0(i)}_{\ell;k}; \hat{y}^{0(i)}_{\ell;k}; \hat{\gamma}^{(i)}_\ell \right\}_{i \in [Q]; \ell \in []; k \in [N]}
$$
$$
\cup \left\{ \hat{b}_{\ell;k;m}; \hat{b}_{\ell;k;m} \right\}_{\ell \in []; m; k \in [N]}
$$

25

5.1. Specication of simulator, S

Let A be a PPT adversary that makes at most $Q = \text{poly}()$ queries to the oracles. The simulator S starts by initializing an empty set of inner products C_{ip} and two empty tables $T_1; T_T$ which map handles to the polynomials over the variables of R. The state of the simulator consists of these three objects, $(C_{ip}; T_1; T_T)$, which are updated after each query received. The simulator S answers the adversary's queries as follows.

Token generation queries. On input $x^{(i)} \in Z_q^n$, O_{TokGen}^0 sends the collection C_{ip}^0 to the simulator.

1. S updates $C_{ip} \quad C_{ip}^C$.
2. For $1 \leq \ell, 1 \leq m \leq N$, S generates a new handle $h_{\ell;m} \xleftarrow{\$} \{0;1\}$ and adds the mapping $h_{\ell;m} \mapsto \Lambda^{(i)} s_{\ell;m}^{(i)}$ to T_1.
3. S then sets $tk_\ell = h_{\ell;1}; \ldots; h_{\ell;N}$.
4. Finally, S returns the token $tk = (tk_1; \ldots; tk)$.

Encryption queries. On input $y^{(i)} \in Z_q^n$,

1. O_{Encrypt}^C sends the collection C_{ip}^C to the simulator.
2. S updates $C_{ip} \quad C_{ip}^C$.
3. For $1 \leq \ell, 1 \leq m \leq N$,
   S generates a new handle $h_{\ell;m} \xleftarrow{\$} \{0;1\}$ and adds the mapping $h_{\ell;m} \mapsto {}^{(\Lambda)} t_{\ell;m} \Lambda^{(i)}$ to T_1.
4. S sets $ct_\ell = h_{\ell;1}; \ldots; h_{\ell;N}$.
5. Finally, S returns the ciphertext $ct = (ct_1; \ldots; ct)$.

Addition oracle queries. Given $h_1; h_2 \in \{0;1\}$, S

1. Veries that formal polynomials $p_1; p_2$ exist in table T, $\in \{1; T\}$ such that $h_1 \mapsto p_1$ and $h_2 \mapsto p_2$. If it is not the case S returns $?$.
2. If a handle for $(p_1 + p_2)$ already exists in T S returns it.
3. Otherwise, S generates a new handle $h \xleftarrow{\$} \{0;1\}$, adds the mapping $h \mapsto (p_1 + p_2)$ to T and returns h.

Pairing oracle queries. Given $h_1; h_2 \in \{0;1\}$, S

1. Veries that formal polynomials $p_1; p_2$ exist in table T_1, such that $h_1 \mapsto p_1$ and $h_2 \mapsto p_2$ in T_1.
2. If it is not the case S returns $?$.
3. If a handle for $(p_1 p_2)$ already exists in T_T, S returns it.
4. Otherwise, S generates a new handle $h \xleftarrow{\$} \{0;1\}$, adds the mapping $h \mapsto (p_1 p_2)$ to T_T and returns h.

Zero-testing oracle queries. Given $h \in \{0, 1\}$, S verifies that formal polynomials p exists in T, $\tau \in \{1, T\}$, such that $h \mapsto \tau p$. If it is not the case S returns $?$. S then works as follows.

1. It "canonicalizes" the polynomial p by expressing it as a sum of products of formal variables in T with $\mathsf{poly}()$ terms.

2. If $\tau = 1$ and p is the zero polynomial, S outputs "zero." Otherwise output "non-zero".

3. If $\tau = T$ the simulator decomposes p into the form

$$
p = \sum_{i,j=1}^{Q} \wedge^{(i)} \wedge^{(j)} \left(p_{i;j} \left(s^{(i)}_{\ell;m}; t^{(j)}_{\ell;m} \right)_{\ell \in 2[]; m \in 2[N]} \right.
$$
$$
\left. + f_{i;j} \left(s^{(i)}_{\ell;m}; t^{(j)}_{\ell;m} \right)_{\ell \in 2[]; m \in 2[N]} \right) \tag{3}
$$

where for $1 \le i, j \le Q$, $p_{i;j}$ is defined as

$$
p_{i;j} = c_{i;j} \left(\sum_{\ell;m=1}^{N} s^{(i)}_{\ell;m} t^{\wedge(j)}_{\ell;m} \right)
$$

where $c_{i;j} \in Z_q$ is the coefficient of the term $s^{(i)}_{1;1} t^{(j)}_{1;1}$ and $f_{i;j}$ consists of the remaining terms.

4. If for all $1 \le i, j \le Q$, $\langle i; j \rangle = 0$ in C_{ip} (corresponding to a zero inner product) and $f_{i;j}$ does not contain any non-zero term, S outputs "zero". Otherwise it outputs "non-zero".

5.2. Correctness of S

Canonicalization is efficient. We first need to show that the canonicalization process in step 1 of Zero-testing oracle queries is efficient. First, note that variables in R are ultimately written as variables in the elements of the underlying matrices B_ℓ and their inverses, which can be tracked efficiently as each entry in the inverse is polynomial size. Since the adversary can only obtain handles to new monomials using token generation and encryption queries, the monomials are all over formal variables in R. Also, since the adversary can make Q queries at most, the polynomial p they can build and submit to the zero-testing oracle has at most $\mathsf{poly}(Q)$ terms and degree 2.

Then using Equations 1 and 2, the formal polynomial p can be expressed as a polynomial over formal variables in T. Since p has degree at most 2 over variables in R, it can be expressed as a sum of at most $\mathsf{poly}(Q; n)$ monomials over variables in T and has degree at most $\mathsf{poly}(n)$. Since both the polynomial over R and the canonical polynomial over T are polynomially-sized, this is efficient.

Correctness of token, encryption, and group queries. The simulator's responses to token generation, encryption and group oracle queries are distributed identically as in the real experiment.

27

Correctness of zero-test queries. We now show correctness of the simulator's answers to zero-testing oracle queries. In this argument we formalize several claims made in Kim et al.'s proof [16]. These statements are not separate claims in Kim et al. so we refer to the page of the ePrint [17] where the claim is made. Unlike prior work we use symmetric bilinear groups. This means, we must argue, that the simulator is correct with the additional exibility provided to the adversary by the ability to take linear combinations of TokGen and Encrypt which are now both in G_1 and to pair these elements. Concretely, this means that the adversary has the ability to ask to pair elements of tk with other elements of tk and elements of ct with elements of ct which was not possible before. In the asymmetric group setting, the adversary was limited to pairing elements in ct with elements in tk. Equation 3 shows how the simulator splits each query into two parts $p_{i;j}$ which consists of valid decryptions (scaled by some values) and $f_{i;j}$ which consist of some other elements. The goal of the proof is to show that for the polynomial $f_{i;j}$, the following two points hold:

1. The terms of $f_{i;j}$ are low degree polynomials of the hidden variables B, and the values $;;$. The polynomial $f_{i;j}$ is either 0 for all values of $x;y$ encrypted by the adversary or non-zero across all values of $x;y$.

2. That the polynomials are low-degree enough that we can use the Schwartz-Zippel lemma (Lemma 1) to show that the nonzero polynomial $f_{i;j}$ evaluates to 0 with low probability. The probability space is the hidden randomness of the scheme, specically the choice of B and the values $;;$.

3. The newly available monomials created by the use of a symmetric bilinear pairing have a dierent degree than the monomials available by properly pairing between tokens and ciphertexts, so we do not have to worry about these canceling each other.

Lemma 2. For $= 1$ the simulator's behavior is correct with overwhelming probability.

Proof of Lemma 2. Note that the only monomials that the adversary obtains are in response to key generation and ciphertext queries. The canonical polynomial

is of the form

$$p = \sum_{i=1}^{Q} \Lambda^{(i)} \left(\sum_{\ell;m=1}^{N} c^{(i)}_{\ell;m;1} s^{(i)}_{\ell;m} + \Lambda^{(i)} \left(\sum_{\ell;m=1}^{N} c^{(i)}_{\ell;m;2} t^{(i)}_{\ell;m} \right) \right) \tag{4}$$

$$= \sum_{i=1}^{Q} \Lambda^{(i)} \left(\sum_{\ell;m=1}^{N} c^{(i)}_{\ell;m;1} \sum_{k=1}^{N} y^{0(i)}_{\ell;k} b^{\wedge}_{;k;m} \right.$$

$$+ \Lambda^{(i)} \left(\sum_{\ell;m=1}^{N} c^{(i)}_{\ell;m;2} \sum_{k=1}^{N} x^{0(i)}_{\ell;k} b^{\wedge}_{;k;m} \right)$$

$$= \sum_{i=1}^{Q} \Lambda^{(i)} \left(\sum_{\ell;m=1}^{N} c^{(i)}_{\ell;m;1} \left(b_{\ell;1;m} + \sum_{k=2}^{N} y^{(i)}_{\ell;k} b^{\wedge}_{;k;m} \right) \right.$$

$$+ \Lambda^{(i)} \left(\sum_{\ell;m=1}^{N} c^{(i)}_{\ell;m;2} \left(b^{\wedge}_{\ell;1;m} + \sum_{k=2}^{N} x^{(i)}_{\ell;k} b^{\wedge}_{\ell;k;m} \right) \right)$$

where the variables

$$\left\{ c^{(i)}_{\ell;m;1}; c^{(i)}_{\ell;m;2} \right\}_{1 \leq \ell,m \leq N} \in Z_q:$$

Note that the sums

$$b_{\ell;1;m} + \sum_{k=2}^{N} y^{(i)}_{\ell;k} b^{\wedge}_{;k;m}$$

can not be the identically zero polynomial over the formal variables

$$f b^{\wedge}_{\ell;k;m} g_{\ell \in [], k;m \in [N]}:$$

The sums

$$b^{\wedge}_{\ell;1;m} + \sum_{k=2}^{N} x_{\ell;k} b^{\wedge}_{\ell;k;m}$$

can only be the identically zero polynomial over the formal variables

$$f b^{\wedge}_{\ell;k;m} g_{\ell \in [], k;m \in [N]}$$

if $^{(i)}_{\ell} = 0$ which happens with negligible probability. Both of these facts are true regardless of the actual values of the adversary's queries. Recall $f \Lambda^{(i)} g_{i \in [Q]}$, $f^{(i)} g_{i \in [Q]}$, and $f b^{\wedge}_{\ell;k;m} g_{\ell \in []; k;m \in [N]}$ are sampled uniformly and independently in the real game. Furthermore, the values $f b^{\wedge}_{\ell;k;m} g_{\ell \in []; k;m \in [N]}$ in the real game are products formed by the inverse computation which are the sum of monomials of degree N. Thus, under the assumption that the above sums are nonzero, the entire value of p can be expressed as a nonzero polynomial of degree at most

29

$N + 1 = \text{poly}()$ in $;;b$. By Lemma 1 (Schwartz-Zippel), p evaluates to non-zero with overwhelming probability for random $;;b$. This implies that the simulator is correct with overwhelming probability. This completes the proof of Lemma 2.

□

Lemma 3. For $= T$ the simulator's behavior is correct with overwhelming probability.

Proof of Lemma 3. We prove Lemma 3 by two claims Claims 1 and 2 that consider whether $f_{i;j}$ has any non-zero terms.

Claim 1. If $f_{i;j}$ does not contain any non-zero term then S outputs the correct value.

Proof of Claim 1. If $f_{i;j}$ does not contain any term, then p is of the form

$$
\begin{aligned}
p &= \sum_{i;j=1}^{Q} \Lambda^{(i)(\Lambda)} \, c_{i;j} \left(\sum_{\prime;m=1}^{;N} \mathfrak{s}^{(i)}_{\prime;m} \mathfrak{t}^{(j)}_{\prime;m} \right)! \\
&= \sum_{i;j=1}^{Q} \Lambda^{(i)(\Lambda)} \, c_{i;j} \left(\sum_{\prime;m=1}^{N} \left(\sum_{k=1}^{N} \mathcal{y}^{0(i)}_{\prime;k} \, b^{\wedge}_{;k;m} \right)! \left(\sum_{k=1}^{N} \mathcal{x}^{0(j)}_{\prime;k} \, b^{\wedge}_{;k;m} \right)! \right)! \\
&= \sum_{i;j=1}^{Q} \Lambda^{(i)(\Lambda)} \, c_{i;j} \sum \left((\mathcal{x}^{0(j)}_{\prime})^{T} \, B \, B^{T} \, \mathcal{y}^{0(i)}_{\prime} \right)^{\prime=1}! \\
&= \sum_{i;j=1}^{Q} \Lambda^{(i)(\Lambda)} \, c_{i;j} \left(\sum_{\prime=1} \langle{}^{(i)}_{\prime} + h \mathcal{x}^{(j)}_{\prime}; \mathcal{y}^{(i)}_{\prime} \rangle \right)! \\
&= \sum_{i;j=1}^{Q} \Lambda^{(i)(\Lambda)} \, c_{i;j} \, h \langle \mathcal{x}^{(j)}_{\prime}; \mathcal{y}^{(i)}_{\prime} \rangle
\end{aligned}
$$

p is the zero polynomial when all $(i;j)$ inner products are zero, which can be known by checking if $(i;j) ! 0$ in C_{ip}. This completes the proof of Claim 1. □

Claim 2. If $f_{i;j}$ has at least one nonzero term then S's output is correct with overwhelming probability.

Now suppose that for some $i;j \in [Q]$ the polynomial $f_{i;j}$ contains at least one term. Then we claim that $f_{i;j}$ cannot be the identically zero polynomial over the formal variables $f \, b_{\prime;k;m} \, g_{\prime 2[];k;m2[N]}$, irrespective of the adversary's choice of admissible queries. To show this we first need to describe all the possible types of terms in $f_{i;j}$. This is a generalization of [16, Lemma 3.3] to the symmetric pairing setting.

Proposition 1. The polynomial $f_{i;j}$ must contain

Cross-Terms A \cross-term" of the form $c \, \mathfrak{s}^{i()}_{;m \, i} \, t^{j()}_{;m \, j}$ where $c \in Z_q$ is non-zero and $(\prime_i; m_i) = (\prime_j; m_j)$.

Square Terms A "square-term" of the form $c \cdot s^{(i)}_{\cdot, i} s^{(j)}_{\cdot, j}$ or $c \cdot t^{(i)}_{\cdot, i} t^{(j)}_{\cdot, j}$ where $c \in Z_q$ is non-zero.

Partial Inner Products Some "partial inner product" of the form $c \cdot s^{(i)}_{\ell, k} t^{(\lambda)}_{\ell, k}$ where $c \in Z_q$ and $\ell \in []; k \in [n]$.

Basic Polynomial No cross-terms, square-terms, or partial inner products and must be a polynomial of only $s^{(i)}_{\cdot}$ or $t^{(j)}_{\cdot}$.

We must show that in the 4 cases, $f_{i;j}$ cannot be identically zero. If the polynomials are nonzero with a degree polynomial in the security parameter, they are unlikely to evaluate to 0.

Note that in all settings we need to consider the fact that the $f_{i;j}$ can contain terms of the form $c \cdot s^{(i)}_{\cdot, i} s^{(j)}_{\cdot, j}$ or $c \cdot t^{(i)}_{\cdot, i} t^{(j)}_{\cdot, j}$ where $c \in Z_q$ is non-zero. Note in the above that $\ell_i; \ell_j$ may be the same or different. The goal is to show that $f_{i;j}$ cannot be identically zero in each of the above cases regardless of the adversary's choice of $\{x^{(i)}; y^{(j)}\}$.

The intuition for this part of the proof is that s terms are formal variables in B_ℓ and are thus degree 1 polynomials of those values. However, B_{ℓ} is an inverse of B_ℓ and thus each entry of B_{ℓ} is a degree N polynomial in the entries of B_ℓ. Recall that s terms are formed from B_ℓ and t terms are formed from B_{ℓ}. This means that cross-terms and square terms are distinct polynomials with degree $N + 1, 2$ (for s cross terms) and $2N$ (for t cross terms). Thus, they never cancel each other.

Cross-Terms and Square Terms. In this case there is some cross term $c \cdot s^{(i)}_{\ell_i, m}$ $t^{(j)}_{\ell_j, m_j}$ where $c \in Z_q$ is non-zero and $(\ell_i; m_i) = (\ell_j; m_j)$. The cross-terms values $c \cdot s^{(i)}_{\ell_i; m_i} t^{(j)}_{\ell_j; m_j}$ were constructed by strictly multiplying elements of B_ℓ with B_{ℓ}^{-1}. Specifically, one can rewrite the above as

$$c \cdot s^{()}_{\ell_i m_i} \, t^{(\lambda)}_{\ell_j m_i} =$$

$$c^{(i)(j)} \left(\hat{b}_{\ell_i; 1; m} + \sum_{k=2}^{N} y^{(i)}_{\ell_i; k} \hat{b}_{\ell_i; k; m} \right) \left(\hat{b}^{(\lambda)}_{\ell_j} \hat{b}_{\ell_j; 1; m_j} + \sum_{k=2}^{N} x^{(j)}_{\ell_j; k-1} \hat{b}_{\ell_j; k; m_j} \right) :$$
(5)

We now recall the form of $\hat{b}$ terms. For a value $k \in [N]$, let $S_{:k}$ denote $[N] \setminus k$ and let $\prod_{k;\ell}$ denote the set of bijections from $S_{:k} \to S_{:\ell}$, and sgn represent a sign function that maps inputs to $\{-1; 1\}$. The variable

$$\hat{b}_{\ell_i; k; m_i} = \frac{\sum_{\prod_{k;m_i}} \text{sgn}(\pi; k; m_i) \prod_{r \in S_{:k}} b_{\ell_i; r; \pi(r)}}{\det(B)} :$$
(6)

Consider the expansion of Equation 5 into monomials of $\hat{b}$. Each monomial in the expansion contains the product of exactly two variables in column ℓ_i of B but no variables in column ℓ_j. However, we now evaluate whether $f_{i;j}$

can be identically zero should these cross-terms include polynomials multiples elements from B' together and/or elements of B'^1 together. First, we must show that these newly formed cross-terms do not contain terms that would cancel with terms from previously constructed cross-terms. This can be shown by evaluating the degrees of the polynomials. All of the monomials in numerator of the denition of Equation 5 have degree exactly N, the new polynomials available to the adversary do not. From Kim et al. [16], we obtain the following claim [17, Page 17, Case 1]. ~~(We refer the reader to [16] for why dierent cross terms cannot cancel one another. Here we focus on why the newly available terms cannot cancel any cross term.)~~

Claim 3. Non-zero \cross-terms" of the form $c\ s^{(i)}_{'_i;m_i}\ t^{(j)}_{'_j;m_j}$, with $('_i;m_i) = ('_j;m_j)$, cannot cancel one another, irrespective of the adversary's choice of admissible queries.

We can then focus on why the newly available terms cannot cancel any cross term. Specically, all terms of the form $c\ s^{(i)}_{'_i}s^{(j)}_{'_j}$ consist of monomials of degree exactly 2. All terms of the form $c\ t^{(i)}_{'_i}t^{(j)}_{'_j}$ consist of monomials of total degree exactly $2N$ (see Equation 6). Therefore, if you have a combination of all these types of cross terms, the resulting polynomial $f_{i;j}$ could not be identically zero.

Basic and Square Terms. We consider the case where $f_{i;j}$ consists of nonzero terms that have no products between s and t. Kim et al. [16] showed non-zero terms of the form $s^{(i)}_{';m}$ or $t^{(j)}_{';m}$ will not cancel out with each other in $f_{i;j}$. We formalize this with the following claim [17, Page 18, Case 3].

Claim 4. Non-zero terms of the form $s^{(i)}_{';m}$ or $t^{(i)}_{';m}$ consisting of monomials of degree 1 and N respectively, are linearly independent and cannot cancel one another, irrespective of the adversary's choice of admissible queries.

However, we now need to consider the case in which we have squared terms. Consider the canonical polynomial in Lemma 2 as described in Equation 4 where there are additionally square terms. That is,

$$p = \sum_{i=1}^{Q} {}^{(i)} \left(\sum_{';m=1}^{N} c^{(i)}_{';m;1}\ s^{(i)}_{';m} \right) + ({}^{(i)})^2 \left(\sum_{';m=1}^{N} c^{(i)}_{';m;3}\ (s^{(i)}_{';m})^2 \right)$$

$$+ {}^{()} \left(\sum_{';m=1}^{N} c^{(i)}_{';m;3}\ t^{(i)}_{';m} \right) + ({}^{()})^2 \left(\sum_{';m=1}^{N} c^{(i)}_{';m;4}\ t^{(i)}_{';m} \right) \tag{7}$$

Recall, that the expansion of t terms are monomials of degree exactly N as described in Equation 6. Thus, note that terms of the type s have degree exactly 1, terms of the type s^2 have degree exactly 2, terms of the type t have degree exactly N and terms of the type t^2 have degree exactly $2N$. Thus, expanding Equation 7 in terms of b yields a polynomial that is not identically

zero and whose monomials are linearly independent. So since there is at least one nonzero coecient the resulting polynomial $f_{i;j}$ could not be identically zero.

Partial Inner Product. In this case there is some partial inner product $c \cdot s_{\cdot;k}^{(i)} \cdot t_{\cdot;k}^{(j)}$ where $c \in Z_q$ and $' \in [];k \in [n]$, but no cross or square terms. From Kim et al. [16], we obtain the following claim [17, Page 18, Case 2]. ~~Kim et al. [16] showed that no form of this term $s_{l;k}^{(i)} t_{l;k}^{(j)}$ consists of monomials of degree exactly N.~~

Claim 5. In $f_{i;j}$, non-zero terms of the form $s_{\cdot;k}^{(i)} t_{\cdot;k}^{(j)}$ cannot cancel one another, irrespective of the adversary's choice of admissible queries.

As before, these terms (of degree $N+1$) will not cancel with the terms available to the adversary which consist of monomials of degree exactly $1, N$ ~~$N + 1$,~~ 2 and $2N$. Therefore, if you have a combination of partial inner products, the resulting polynomial $f_{i;j}$ could not be identically zero.

~~Basic Terms This argument is identical to the case of Kim et al. [16]. In particular, the terms of $f_{i;j}$ must be linearly independent and $f_{i;j}$ is not identically zero.~~

Since the polynomial $f_{i;j}$ has non-zero degree it is not identically zero ~~Since we have shown that the polynomial $f_{i;j}$ can not be identically zero~~ and it has polynomial degree in all cases, the simulator's output of \non-zero" is correct with overwhelming probability by the Schwartz-Zippel lemma (Lemma 1). This completes the proof of Lemma 3. □

This completes the proof of Theorem 1.

5.3. Limitation of the multirandom projection technique

The technique presented in this section requires to modify the internal workings of the underlying IPE scheme. Thus it is not a black-box technique.

To give an intuition on why it cannot be applied in a black-box manner we'll consider Barbosa et al.'s predicate IPE scheme. Ciphertexts and tokens in Barbosa et al.'s scheme are of the form

$$ct_x = [\ x^\top\ B]_1$$

and

$$tk_y = [\ y^\top\ B]_2$$

where ; are independently and randomly sampled in Z_q for each ciphertext and token. Decryption yields $[\ x^\top\ B\ B^\top\ y]_T$. Since $\langle x; y \rangle$ is a multiplicative factor there, can be ignored when the inner product evaluates to zero.

If we tried to apply our technique in a black box manner, these random values would be preserved. Then each sub-ciphertext and token would have independent randomness from the underlying IPE scheme. During decryption, this would cause an issue as we would not be able to factorize the randomness (since it was sampled independently) and thus the sum of the inner products

would not cancel out when needed. We would indeed end up with something of the form: $r_1 \langle x_1; y_1 \rangle + \cdots + r_\ell \langle x_\ell; y_\ell \rangle$, which would not evaluate to zero when the partial inner products are not zero (with high probability), even though the overall inner product $\langle x; y \rangle$ is itself zero. We thus need to remove this internal randomness and replace it by a global one. □

6. Building distance hiding PSE

As mentioned in Section 2, Hamming distance can be calculated using the inner product between the two biometric vectors. As such, we can use a range of possible inner product values as the distance threshold.

Predicate function-hiding secret key IPE [52], or $IPE_{fh;sk;pred}$, allows one to test if the inner product between two vectors is equal to zero. By appending a value to the rst vector and -1 to the second vector, we can support equality testing for non-zero values. Generating several tokens or ciphertexts, one per distance in the range, allows to test if the inner product is below the chosen threshold.

We show that one can use $IPE_{fh;sk;pred}$ to construct PSE for Hamming distance[14]. At a high level, each keyword is encoded as a $\{-1; 1\}$ vector and -1 is appended to it, which in turn is encrypted with $IPE_{fh;sk;pred}$. Keywords are similarly encoded but this time a distance from the range is appended to them, and tokens generated as part of the underlying $IPE_{fh;sk;pred}$ scheme.

Construction 1 (Proximity Searchable Encryption). Fix the security parameter $\lambda \in \mathbb{N}$. Let $IPE_{fh;sk;pred}$ = (IPE:Setup, IPE:TokGen, IPE:Encrypt; IPE:Decrypt) be a predicate function-hiding secret key IPE scheme over $\mathbb{Z}_q^{n+1}$. Let $x_i \in \mathbb{Z}_q^n$ and $X = (x_1; \cdots ; x_\ell)$ be the list of keywords. Let F be the set of all predicates such that for any $x_i \in X$, $f_{y;t}(x_i) = 1$ if the Hamming distance between x_i and the query vector $y \in \mathbb{Z}_q^n$ is less or equal to some chosen threshold $t \in \mathbb{Z}_q$, $f_{y;t}(x_i) = 0$ otherwise. Figure 6 is a proximity searchable encryption scheme for the Hamming distance.

Theorem 2 (PSE main theorem). Let $IPE_{fh;sk;pred}$ = (IPE:Setup, IPE:TokGen, IPE:Encrypt, IPE:Decrypt) be an IND-secure function-hiding inner product predicate encryption scheme over $\mathbb{Z}_q^{n+1}$. Then $\exists PSE$ = (PSE:Setup, PSE:BIndex, PSE:Trpdr, PSE:Search), a secure proximity searchable encryption scheme for the Hamming distance, such that for any PPT adversary A_{PSE} for Exp_{IND}^{PSE}, there exists a PPT adversary A_{IPE} for Exp_{IND}^{IPE}, such that for any security parameter $\lambda \in \mathbb{N}$,

$$Adv_{A_{PSE}}^{Exp_{IND}^{PSE}} = Adv_{A_{IPE}}^{Exp_{IND}^{IPE}}$$

[14]Support of addition/deletion of records seems achievable by deleting after search and inserting new ciphertexts in the database. However this would result in additional access pattern leakage since these record would be clearly identiable by the server.

Proof of Theorem 2. The correctness of the scheme follows from the correctness of the underlying IPE scheme. Assume there exists $x_i \in X$, $i \in [1; \ell]$, such that $f_{y;t}(x_i) = 1$. That is $D(y; x_i) \leq t$ with $D(y; x_i)$ the Hamming distance between vectors y and x_i. Then there exists a unique $tk_j \in Q_{y;t}$ such that $b_j \leftarrow$ IPE:Decrypt$(pp; tk_j; ct_i)$ and $b = 1$ with overwhelming probability by the correctness of the IPE scheme. Now assume that for some $x_i \in X$, $i \in [1; \ell]$, we have $f_{y;t}(x_i) = 0$. Then for all $tk_j \in Q_{y;t}$, $b_j \leftarrow$ IPE:Decrypt$(pp; tk_j; ct_i)$ and $b_j = 1$ with negligible probability. Then considering the worst case where either $D(y; x\cdot) = t$ or for all $x_i \in X$, $f_{y;t}(x_i) = 0$, we have:

$$\Pr[\text{PSE:Search}(pp; Q_{y;t}; I_X) = J_{X;y;t}]$$

$$\geq 1 - \ell(t+1) \Pr\left[\text{IPE:Decrypt}(pp; tk_j; ct_i) \neq \prod_i h^{(D(x_i;y) \stackrel{?}{=} d_j)}\right]$$

$$\geq 1 - \ell(t+1) \text{negl}():$$

We now prove the security of the construction. Let A_{PSE} be a PPT adversary for the experiment Exp^{PSE}_{IND} and C_{IPE} be an challenger for Exp^{IPE}_{IND}. We build a PPT adversary A_{IPE} for the experiment Exp^{IPE}_{IND} as follows:

1. A_{IPE} receives pp from C_{IPE} and forwards it to A_{PSE}.
2. A_{IPE} receives two m-query histories $\text{History}^{(0)}; \text{History}^{(1)}$ from A_{PSE} where $\text{History}^{(\beta)} = (X^{(\beta)}; F^{(\beta)})$ for $\beta \in \{0; 1\}$.
3. For each $x_i^{(\beta)} \in X^{(\beta)}$, $i \in [1; \ell]$, A_{IPE} encodes it as $x_i^{(\beta)} \in \{-1; 1\}^n$ and creates the query $S_i = (x_i^{(0)}\|j-1; x_i^{(1)}\|j-1):$
4. A_{IPE} sets $S = S_1; \cdots; S_\ell$.
5. For each $f_j^{(\beta)} \in F^{(\beta)}$, $j \in [1; m]$:

   (a) A_{IPE} extracts a vector $y_j^{(\beta)} \in Z_q^n$ and $t \in N$.

   (b) A_{IPE} encodes $y_j^{(\beta)}$ as $y_j^{(\beta)} \in \{-1; 1\}^n$ and creates $D_j^{(0)} = (d_0; \cdots; d_t)$ such that $d_k = n - 2k$ with $0 \leq k \leq t$.

   (c) A_{IPE} creates $D_j^{(0)}$ by reordering the elements in $D_j^{(0)}$ such that for all $k \in [0; t]$ and $d_k^{(0)} \in D_j^{(0)}; d_k^{(1)} \in D_j^{(1)}$ we have $\langle hx_i^{(0)}; y_j^{(0)}i \stackrel{?}{=} d_k^{(0)} = \langle hx_i^{(1)}; y_j^{(1)}i \stackrel{?}{=} d_k^{(1)}$: (A_{IPE} can always find a permutation to make this last condition by the admissibility requirement.)

   (d) A_{IPE} samples a random permutation $\pi_j : [0; t] \to [0; t]$.

   (e) For $0 \leq k \leq t$, A_{IPE} creates $y_j^{(\beta)}\|d^{(\beta)}$ with $\beta \in \{0; 1\}$, $d^{(0)} \in D_k^{(0)}$ and $d_j^{(1)} \in D_k^{(1)}$. Then A_{IPE} computes

   $$R_j^{(\beta)} = \left(\pi_j\left(y_j^{(\beta)}\|d^{(\beta)}\right); \cdots; y_j^{(\beta)}\|d^{(\beta)}\right)_{j=0}^{t}$$

   and sets $R_j = (R_j^{(0)}; R_j^{(1)})$.

   (f) A_{IPE} sets $R = R_1; \cdots; R_m$.

	Underlying IPE scheme			
	MRProj	[44, Section 4]	[44, Section 5]	[82]
Setup	$((n+1)=)^3$	$(n+1)^3$	$(n+1)^3$	$(6n+6)^3$
BIndex	$'(n+ +1)$	$'(n+1)$	$'(12n+21)$	$6'(n+1)$
Trpdr	$(t+1)(n+ +1)$	$(t+1)(n+1)$	$(t+1)(12n+21)$	$6(t+1)(n+1)$
Search	$'(t+1)(n+ +1)$	$'(t+1)(n+1)$	$'(t+1)(6n+12)$	$6'(t+1)(n+1)$
jskj	$2((n+1)^2= +2n+ +3)$	$2(n+1)^2+2$	$24n+42$	$60(n+1)^2$
jlj	$'(n+ +1)$	$'(n+1)$	$'(6n+12)$	$6'(n+1)$
$jtk_{y;t}j$	$(t+1)(n+ +1)$	$(t+1)(n+1)$	$(t+1)(6n+12)$	$6(t+1)(n+1)$

Table 5: PSE scheme eciency for keywords of size n depending on underlying $IPE_{fh;sk;pred}$ scheme. Upper part of the table shows number of group or pairing operations per function. Lower part of the table shows number of group elements per component. The number n is the length of the biometric template, is the number of bases in the multi random projection scheme, t is the desired distance tolerance, and ' is the total number of records in the database.

6. A_{IPE} sends the token generation queries R and encryption queries S to C_{IPE} and receives back a set of tokens $T^{()} = tk^{()}_{1;o}$; $tk^{()}_{m;a}$ and a set of encrypted keywords $C^{()} = ct^{()}_1$; $ct^{()}$ such that

$$tk^{()}_{j;k} \quad IPE:TokGen(sk; \; y^{()}_j jj \; d^{()})_k$$
$$ct^{()}_i \quad IPE:Encrypt(sk; \; x^{()}_i jj \; -1)$$

for i 2 [1; '], j 2 [1; m], k 2 [0; t] and 2 f0; 1g. A_{IPE} forwards $T^{()}$ and $C^{()}$ to A_{PSE}, respectively as the encrypted index $I^{()}$ and the list of queries $Q^{()}$.

7. A_{IPE} receives ^{0} 2 f0; 1g from A_{PSE} and returns it.

Since the number of token generation queries, m t, sent by A_{IPE} remains polynomial in the security parameter, the advantage of A_{PSE} is

$$Adv^{Exp^{PSE}_{IND}}_{A_{PSE}} = Adv^{Exp^{IPE}_{IND}}_{A_{IPE}}$$

This completes the proof of Theorem 2. $\square$

Table 5 presents the resulting eciency of distance hiding PSE schemes based on dierent $IPE_{fh;sk;pred}$ constructions. This table corresponds to t + 1 tokens with all operations on dimension n + 1.

7. Implementation

This section presents an implementation and an evaluation of the PSE scheme proposed in this paper. We implemented the MRProj construction described in section 5 and the resulting PSE (see section 6) schemes in Python 3. These implementations can be found in a Github repository [83]. Our IPE implementations uses the Charm [78] and FLINT [23] libraries for the pairing group operations and nite eld arithmetic in Z_q. For comparison purposes, we used the pairing group over the asymmetric curve MNT159, the same as in Kim et

36

PSE:Setup(1) ! (sk; pp):

   1. Run and output (sk; pp)   IPE:Setup(1).

PSE:Trpdr(sk; $f_{y;t}$) ! $Q_{y;t}$:

   1. For 0 j t compute $d_j = n$ $2j$. 2.
Set $D = (d_0; :::; d_t)$.
   3. Sample random permutation : $[0; t]$! $[0; t]$.
   4. Compute $D = (D) = fd_0; \ ; dg._t$.
   5. Encode y as y $2 f$ $1; 1g^n$.
   6. For 0 j t call tk_j IPE:TokGen(sk; y jj d)$_j$ 7.
Output $Q_{y;t} = (tk_0; \ ; tk_t)$.

PSE:BIndex(sk; X) ! I_X:

   1. For each keyword $x_i 2 X$, $i 2 f1; \ ; 'g$,
     encode $x_i 2 f$ $1; 1g^n$,
     compute ct_i IPE:Encrypt(sk; $x_i$jj -1).
   2. Output $I_X = (ct_1; \ ; ct_')$.

PSE:Search(pp; $Q_{y;t}$; I_X) ! $J_{X;y;t}$:

   1. Initialize $J_{X;y;t} = ;$.
   2. For each $ct_i 2 I_X$ and for each $tk_j 2 Q_{y;t}$,
     call b_j IPE:Decrypt(pp; tk_j; ct_i).
     If $b_j = 1$, add i to $J_{X;y;t}$, continue to ct_{i+1}.
   3. Output $J_{X;y;t}$.

Figure 6: Construction of proximity search from $IPE_{fh;sk;pred}$.

al.'s FHIPE implementation [84]. For testing with a symmetric pairing group we used SS512.

The search, encryption and token generation algorithms were parallelized. Benchmarking tests for each algorithm were implemented and the number of random projections, the distance threshold and the input vector sizes for these tests can vary. This allowed us to compare eciency for dierent parameters and pinpoint values that yield a practical and accurate scheme. With a number of random projections equal to 1, we obtain Setup timings and secret key size for RProjC. Setting the distance threshold to 0 allows us to get timings for MRProjC. We used iris readings from the ND 0405 as input vectors to the benchmarking tests to be as realistic as possible.

7.1. Evaluation

We evaluate our implementations on a Linux server with an AMD Ryzen 9 3950X 16-Core processor and 64GB of RAM. Remember that the preferred input vector size for correctness is 1024 (as stated in Section 4).

n		t 128	Time							
			MRProj w/ SS512			MRProj w/ MNT159			MRProjC	
			BIndex	Trpdr	Search	BIndex	Trpdr	Search	Trpdr	Search
3	38 192	5	.7	.8	58	1.5	.36	234	.01	31
	57 256	7	.9	1.8	130	2.2	.8	495	.01	46
	76 384	10	1.2	3.2	228	2.9	1.4	850	.02	62
	115 512	13	1.7	7.4	514	4.4	3.1	1870	.03	92
	153 768	19	2.3	13	907	5.7	5.7	3282	.04	140
	230 1024	25	3.4	28	2030	8.6	13.4	7210	.06	185
		307	4.3	52	3580	10.8	22.4	12600	.08	241

Table 6: Operations timing (in seconds) for dierent vector sizes. n is the vector length, the number of bases used, and t = :30 the distance tolerance. BIndex procedure for MRProj and MRProjC schemes are the same procedures, MRProjC uses vectors whose length is 1 fewer. We only report timing for BIndex for MRProj.

n		t	Time		
			MRProj w/ SS512	MRProj w/ MNT159	RProjC
128	3	38	34	75	4 10^3
192	5	57	38	47	1:3 10^4
256	7	76	43	57	3:2 10^4
384	10	115	73	94	1:1 10^5
512	13	153	106	153	2:6 10^5
768	19	230	169	269	8:6 10^5
1024	25	307	225	268	2:0 10^6

Table 7: Operations timing in seconds for Setup dierent vector sizes. n is the vector length, the number of bases used, and t = :30 the distance tolerance. Setup for MRProj and MRProjC schemes are the same procedures, MRProjC uses vectors whose length is 1 fewer. We only report Setup for MRProj. Timing for MRProjC Setup is interpolated. Measured n = 10 to 240 in steps of 10 cubic t with coecients y = :003x^3 :578x^2 + 36x 557 with R^2 = :996.

Timing. We evaluate the timing eciency of our PSE construction with and without the multi random projection technique. When using the multi random projection technique we report on timings for both the asymmetric MNT159 curve and the symmetric SS512 curve. Table 7 reports the timings for Setup and table 6 reports timings for the other algorithms of the PSE scheme. RProjC corresponds to Kim et al.'s FHIPE construction. MRProjC corresponds to the same scheme but with the multi random projection technique applied. In the last column of the timing section of the table, we report the timing of the Setup algorithm without this multi random projection construction.

During our tests, we noticed a jump in Setup timings when going from sub-vectors of 40 to 60 group elements, we thus chose values that yield sub-vectors lengths of approximately 40. We make four main observations.

1. Setup and BIndex have comparable performance for MRProj and MRProjC (the only dierence is adding 1 to underlying dimension). However, Trpdr is substantially slower for MRProj since it prepares t + 1 tokens, but performance remains reasonable.
2. Distance hiding has a large impact on the Search algorithm. MRProjC

			Sizes		
			MRProj		RProjC [16]
n		t	jEncDBj	jskj	jskj
128	3	38	5:9 MB	560 KB	1:6 MB
192	5	57	8:9 MB	770 KB	3:6 MB
256	7	76	12 MB	980 KB	6:4 MB
384	10	115	18 MB	1:5 MB	14 MB
512	13	153	24 MB	2:1 MB	26 MB
768	19	230	36 MB	3:2 MB	57 MB
1024	25	307	47 MB	4:3 MB	100 MB

Table 8: Sizes (in Megabytes/Kilobytes) for dierent vector sizes. n is the vector length, the number of bases used, and t = :30 the distance tolerance. Storage for the MRProjC Setup is interpolated. Measured n = 10 to 240 in steps of 10 quadratic t with coecients y = $96x^2$ + 192x + 573 with R^2 = 1.

Search takes 4 minutes, MRProj Search takes 3.5 hours for the MNT159 curve and 1 hour for the SS512 curve. All approaches scan the whole database which is problematic for large datasets. We discuss possible solutions in Section 8.

3. The use of a symmetric pairing dramatically improves search time by roughly a factor of between 3 and 4 across testing parameters. However, it does increase the time to generate the trapdoor by roughly a factor of 2.

4. Finally, this table shows that Setup without multi random projection is completely impractical for large input vector sizes. In particular, for vectors of size 1024, Setup takes approximately 23 days. In comparison, Setup using multi random projection takes less than ve minutes for input vec-tors of size 1024. Our multi random projection construction thus allows to use a large enough input vector size to maintain a high correctness while increasing the eciency of the setup algorithm. This is explained by the fact that the Setup algorithm's running time is dominated by the matrix inversion. It is then more ecient to perform multiple inversions of small matrices than a single inversion of a bigger one.

Storage. Table 8 reports on the sizes of the encrypted database and secret key. We evaluate the impact of the multi random projection PSE construction on storage eciency. As can be seen on Table 8, the impact is low for small input vectors, however, it makes a big dierence for larger ones. Indeed, when the size of the [16] grows quadratically with the vector size, the size of the key generated with the multi random projection technique grows with $(n=)^2$ $n^2=$. For vectors of size 1024, we consider = 25 and the secret key generated with the multi random projection technique is 23:2 times smaller than the single basis key, conrming the asymptotic analysis.

8. Conclusion

Iris biometric feature extractors produce feature vectors similar in the binary Hamming metric. Inner product encryption was proposed to build encrypted search for the binary Hamming metric. In this work we explored a domain specic solution for secure searchable encryption for iris biometric databases.

We observed in the statistics of the iris biometric data that large vectors are required for both correctness and minimizing leakage. With large vectors, we see that the distance between readings of the same class can be separated from the distance distribution from the readings of other classes (see Figure 4). This means that with a xed distance threshold, we can ensure that more readings of the same class are approved while readings from other classes are denied (with high probability).

In prior work, Setup was not feasible for large vector lengths due to the cost of inverting large matrices. In the most relevant prior work [16], they skip this step in benchmarking due to the high cost. Our interpolation results show that for $n = 1024$ would take roughly 23 days. This is estimated on a parallel implementation in C. The length $n = 1024$ is the length of prior iris feature extractors. We do not consider this time acceptable.

In the RProjC scheme of Kim et al. [16], additionally the distance is leaked between queries and all points in the database. Based on prior work on trilateriliation, with a constant number of queries observed in n, the server can build complete distance information between the stored data points. If the adversary knows auxiliary information about the database, the encryption may not protect the data at all.

In this work we oer solutions to these two problems. We show a multi random projection approach that allows for breaking large vectors into small vectors. This allows us to use smaller matrices greatly reducing the computational time required to invert the matrices. Doing two $n=2$ inversions takes $1=4$ the time of one size n inversion. Careful optimization improves Setup time by four orders of magnitude while only increasing search time by 3%.

We show how to use predicate inner product encryption to build a scheme that hides the distance between the query and the stored records. By using a predicate scheme instead of one that gives the value of the inner product, the server only learns if the two vectors are a xed distance from one another. This greatly reduces the information that is leaked through remotely executing this operation. The server only learns information about data that are close the queried point and learns nothing about data that are outside the distance threshold. We show this scheme leaks only access pattern and distance equality leakage.

The improvement in accuracy for higher n also yields an improvement of leakage prole for our scheme. When two or more classes are returned from a single query, this leaks that the returned items are within distance $2t$ (through access pattern) and whether they are the same distance from the query (distance equality leakage). Decreasing the statistical overlap between classes minimizes the probability of both leakages which translates to a more private system for

sensitive biometric data.

The transformation comes at a cost of making search slower and no longer appropriate for moderately sized databases. We believe that this transformation is required in order to maintain the integrity of sensitive biometric information. Thus, our main open problem is whether or not this signicant slow down to search is avoidable. For databases at larger scales, doing a linear search of the entire database for each query is unacceptable. With our distance hiding transformation we have to do a linear scan for each subtoken (that checks a specic distance) and so we see a signicant (but linear) slowdown over a single linear database scan. Of particular interest are approaches that use indices that natively support k nearest neighbors but are not vulnerable to recent attacks (such as [33, 34]) and interactive solutions where the client can guide the search.

References

[1] Chloe Cachet, Sohaib Ahmad, Luke Demarest, Ariel Hamlin, and Benjamin Fuller. Proximity searchable encryption for the iris biometric. In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '22, page 1004{1018, New York, NY, USA, 2022. Association for Computing Machinery.

[2] S. Prabhakar, S. Pankanti, and A.K. Jain. Biometric recognition: security and privacy concerns. IEEE Security & Privacy, 1(2):33{42, 2003.

[3] Javier Galbally, Arun Ross, Marta Gomez-Barrero, Julian Fierrez, and Javier Ortega-Garcia. From the iriscode to the iris: A new vulnerability of iris recognition systems. Black Hat Briengs USA, 1, 2012.

[4] Guangcan Mai, Kai Cao, Pong C Yuen, and Anil K Jain. On the reconstruction of face images from deep face templates. IEEE transactions on pattern analysis and machine intelligence, 41(5):1188{1202, 2018.

[5] Sohaib Ahmad and Benjamin Fuller. Resist: Reconstruction of irises from templates. In 2020 IEEE International Joint Conference on Biometrics (IJCB), pages 1{10. IEEE, 2020.

[6] Shreyas Venugopalan and Marios Savvides. How to generate spoofed irises from an iris code template. IEEE Transactions on Information Forensics and Security, 6(2):385{395, 2011.

[7] Yi Huang, Adams Kong Wai-Kin, and Kwok-Yan Lam. From the perspective of CNN to adversarial iris images. In 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), pages 1{10. IEEE, 2018.

[8] Sobhan Soleymani, Ali Dabouei, Jeremy Dawson, and Nasser M Nasrabadi. Adversarial examples to fool iris recognition systems. In 2019 International Conference on Biometrics (ICB), pages 1{8. IEEE, 2019.

[9] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, et al. The matter of heartbleed. In Proceedings of the 2014 conference on internet measurement conference, pages 475{488, 2014.

[10] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. ACM SIGARCH Computer Architecture News, 42(3):361{372, 2014.

[11] Onur Mutlu and Jeremie S Kim. Rowhammer: A retrospective. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(8):1555{1571, 2019.

[12] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000, pages 44{55. IEEE, 2000.

[13] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved denitions and ecient construc-tions. Journal of Computer Security, 19:895{934, 01 2011.

[14] Christoph Bosch, Pieter Hartel, Willem Jonker, and Andreas Peter. A survey of provably secure searchable encryption. ACM Computing Surveys (CSUR), 47(2):1{51, 2014.

[15] Benjamin Fuller, Mayank Varia, Arkady Yerukhimovich, Emily Shen, Ariel Hamlin, Vijay Gadepally, Richard Shay, John Darby Mitchell, and Robert K Cunningham. SoK: Cryptographically protected database search. In 2017 IEEE Symposium on Security and Privacy (SP), pages 172{191. IEEE, 2017.

[16] Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J Wu. Function-hiding inner product encryption is practical. In International Conference on Security and Cryptography for Networks, pages 544{562. Springer, 2018.

[17] Sam Kim, Kevin Lewi, Avradip Mandal, Hart William Montgomery, Arnab Roy, and David J Wu. Function-hiding inner product encryption is practical. IACR Cryptology ePrint Archive, 2016:440, 2016.

[18] John Daugman. Results from 200 billion iris cross-comparisons. 01 2005.

[19] John Daugman. How iris recognition works. In The essential guide to image processing, pages 715{739. Elsevier, 2009.

[20] Nadia Othman, Bernadette Dorizzi, and Sonia Garcia-Salicetti. Osiris: An open source iris recognition software. Pattern Recognition Letters, 82:124{131, 2016.

[21] Sohaib Ahmad and Benjamin Fuller. Thirdeye: Triplet-based iris recognition without normalization. In IEEE International Conference on Biometrics: Theory, Applications and Systems, 2019.

[22] Tatsuaki Okamoto and Katsuyuki Takashima. Dual pairing vector spaces and their applications. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 98(1):3{15, 2015.

[23] William B. Hart. Fast library for number theory: An introduction. In Komei Fukuda, Joris van der Hoeven, Michael Joswig, and Nobuki Takayama, editors, Mathematical Software { ICMS 2010, pages 88{91, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[24] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In Advances in Cryptology { EUROCRYPT 2012, pages 591{608, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[25] Sohaib Ahmad, Chloe Cachet, Luke Demarest, Benjamin Fuller, and Ariel Hamlin. Proximity searchable encryption for the iris biometrics. Cryptology ePrint Archive, Report 2020/1174, 2020. https://ia.cr/2020/1174.

[26] P Jonathon Phillips, W Todd Scruggs, Alice J O'Toole, Patrick J Flynn, Kevin W Bowyer, Cathy L Schott, and Matthew Sharpe. FRVT 2006 and ICE 2006 large-scale experimental results. IEEE transactions on pattern analysis and machine intelligence, 32(5):831{846, 2009.

[27] Kevin W Bowyer and Patrick J Flynn. The ND-IRIS-0405 iris image dataset. arXiv preprint arXiv:1606.04853, 2016.

[28] P Jonathon Phillips, Kevin W Bowyer, Patrick J Flynn, Xiaomei Liu, and W Todd Scruggs. The iris challenge evaluation 2005. In 2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems, pages 1{8. IEEE, 2008.

[29] Mehmet Kuzu, Mohammad Saiful Islam, and Murat Kantarcioglu. Ecient similarity search over encrypted data. In 2012 IEEE 28th International Conference on Data Engineering, pages 1156{1167. IEEE, 2012.

[30] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In Proceedings of the thirtieth annual ACM symposium on Theory of computing, pages 604{613, 1998.

[31] Nissanka Bodhi Priyantha, Hari Balakrishnan, Erik D Demaine, and Seth Teller. Mobile-assisted localization in wireless sensor networks. In Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., volume 1, pages 172{183. IEEE, 2005.

[32] James Aspnes, Tolga Eren, David Kiyoshi Goldenberg, A Stephen Morse, Walter Whiteley, Yang Richard Yang, Brian DO Anderson, and Peter N Belhumeur. A theory of network localization. IEEE Transactions on Mobile Computing, 5(12):1663{1678, 2006.

[33] Evgenios M Kornaropoulos, Charalampos Papamanthou, and Roberto Tamassia. Data recovery on encrypted databases with k-nearest neighbor query leakage. In 2019 IEEE Symposium on Security and Privacy (SP), pages 1033{1050. IEEE, 2019.

[34] Evgenios M Kornaropoulos and Petros Efstathopoulos. The case of adversarial inputs for secure similarity approximation protocols. In 2019 IEEE European Symposium on Security and Privacy (EuroS&P), pages 247{262. IEEE, 2019.

[35] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. Access pattern disclosure on searchable encryption: ramication, attack and mitigation. In NDSS, volume 20, page 12. Citeseer, 2012.

[36] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-abuse attacks against searchable encryption. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pages 668{679, 2015.

[37] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'Neill. Generic attacks on secure outsourced databases. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 1329{1340, 2016.

[38] Guofeng Wang, Chuanyi Liu, Yingfei Dong, Hezhong Pan, Peiyi Han, and Binxing Fang. Query recovery attacks on searchable encryption based on partial knowledge. In International Conference on Security and Privacy in Communication Systems, pages 530{549. Springer, 2017.

[39] Paul Grubbs, Kevin Sekniqi, Vincent Bindschaedler, Muhammad Naveed, and Thomas Ristenpart. Leakage-abuse attacks against order-revealing encryption. In Security and Privacy (SP), 2017 IEEE Symposium on, pages 655{672. IEEE, 2017.

[40] Paul Grubbs, Marie-Sarah Lacharite, Brice Minaud, and Kenneth G Paterson. Pump up the volume: Practical database reconstruction from volume leakage on range queries. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pages 315{331, 2018.

[41] Evangelia Anna Markatou and Roberto Tamassia. Full database reconstruction with access and search pattern leakage. In International Conference on Information Security, pages 25{43. Springer, 2019.

[42] Evgenios M Kornaropoulos, Charalampos Papamanthou, and Roberto Tamassia. The state of the uniform: attacks on encrypted databases beyond the uniform query distribution. In 2020 IEEE Symposium on Security and Privacy (SP), pages 1223{1240. IEEE, 2020.

[43] Francesca Falzon, Evangelia Anna Markatou, David Cash, Adam Rivkin, Jesse Stern, and Roberto Tamassia. Full database reconstruction in two dimensions. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pages 443{460, 2020.

[44] Manuel Barbosa, Dario Catalano, Azam Soleimanian, and Bogdan Warinschi. Ecient function-hiding functional encryption: From inner-products to orthogonality. In Mitsuru Matsui, editor, Topics in Cryptology { CT-RSA 2019, pages 127{148, 2019.

[45] Paul Grubbs, Anurag Khandelwal, Marie-Sarah Lacharite, Lloyd Brown, Lucy Li, Rachit Agarwal, and Thomas Ristenpart. Pancake: Frequency smoothing for encrypted data stores. In 29th USENIX Security Symposium, pages 2451{2468, 2020.

[46] Alexandra Boldyreva and Tianxin Tang. Privacy-preserving approximate k-nearest-neighbors search that hides access, query and volume patterns. PoPETS Proceedings on Privacy Enhancing Technologies, 2021.

[47] Jacob T. Schwartz. Probabilistic algorithms for verication of polynomial identities. In Edward W. Ng, editor, Symbolic and Algebraic Computation, pages 200{215, Berlin, Heidelberg, 1979. Springer Berlin Heidelberg.

[48] Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, Symbolic and Algebraic Computation, pages 216{226, Berlin, Heidelberg, 1979. Springer Berlin Heidelberg.

[49] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matt Franklin, editor, Advances in Cryptology { CRYPTO 2004, pages 41{55, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[50] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, Advances in Cryptology { EUROCRYPT 2005, pages 440{456, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[51] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel Smart, editor, Advances in Cryptology { EUROCRYPT 2008, pages 146{162, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[52] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Theory of Cryptography, pages 457{473, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[53] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Ran Canetti and Juan A. Garay, editors, Advances in Cryptology { CRYPTO 2013, pages 461{478, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[54] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private subspace-membership encryption and its applications. In Kazue Sako and Palash Sarkar, editors, Advances in Cryptology - ASIACRYPT 2013, pages 255{275, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[55] Sha Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, Advances in Cryptology { EUROCRYPT 2014, pages 578{602, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[56] Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In Jean-Sebastien Coron and Jesper Buus Nielsen, editors, Advances in Cryptology { EUROCRYPT 2017, pages 601{626, Cham, 2017. Springer International Publishing.

[57] Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In Hovav Shacham and Alexandra Boldyreva, editors, Advances in Cryptology { CRYPTO 2018, pages 597{627, Cham, 2018. Springer International Publishing.

[58] Jin Li, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou. Fuzzy keyword search over encrypted data in cloud computing. In _INFOCOM, 2010 Proceedings IEEE_, pages 1{5. IEEE, 2010.

[59] Jianfeng Wang, Hua Ma, Qiang Tang, Jin Li, Hui Zhu, Siqi Ma, and Xiaofeng Chen. Ecient veriable fuzzy keyword search over encrypted data in cloud computing. _Comput. Sci. Inf. Syst._, 10(2):667{684, 2013.

[60] Alexandra Boldyreva and Nathan Chenette. Ecient fuzzy search on encrypted data. In _International Workshop on Fast Software Encryption_, pages 613{633. Springer, 2014.

[61] Kai Zhou and Jian Ren. Passbio: Privacy-preserving user-centric biometric authentication. _IEEE Transactions on Information Forensics and Security_, 13(12):3050{3063, 2018.

[62] Seny Kamara, Tarik Moataz, and Olya Ohrimenko. Structured encryption and leakage suppression. In _CRYPTO_, pages 339{370. Springer, 2018.

[63] Yupeng Zhang, Jonathan Katz, and Charalampos Papamanthou. All your queries are belong to us: The power of le-injection attacks on searchable encryption. In _25th USENIX Security Symposium_, pages 707{720, 2016.

[64] M. Lacharite, B. Minaud, and K. G. Paterson. Improved reconstruction attacks on encrypted data using range query leakage. In _2018 IEEE Symposium on Security and Privacy (SP)_, pages 297{314, 2018.

[65] Cem Evrendilek and Huseyin Akcan. On the complexity of tion with noisy range measurements. _trilatera-IEEE Communications Letters,_ 15(10):1097{1099, 2011.

[66] Richard C Tillquist, Rafael M Frongillo, and Manuel E Lladser. Metric dimension. _arXiv preprint arXiv:1910.04103_, 2019.

[67] Lucas Laird, Richard C Tillquist, Stephen Becker, and Manuel E Lladser. Resolvability of Hamming graphs. _SIAM Journal on Discrete Mathematics_, 34(4):2063{2081, 2020.

[68] Lucas Laird. Metric dimension of hamming graphs and applications to computational biology. _arXiv preprint arXiv:2007.01337_, 2020.

[69] Karen Hollingsworth, Kevin W Bowyer, and Patrick J Flynn. Similarity of iris texture between identical twins. In _2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops_, pages 22{29. IEEE, 2010.

[70] Sohaib Ahmad and Benjamin Fuller. Unconstrained iris segmentation using convolutional neural networks. In _Asian Conference on Computer Vision_, pages 450{466. Springer, 2018.

[71] Sohaib Ahmad. Sohaib ahmad github, 2020. Accessed: 2020-07-23.

[72] Ajay Kumar and Arun Passi. Comparison and combination of iris matchers for reliable personal authentication. Pattern recognition, 43(3):1016{1026, 2010.

[73] Sailesh Simhadri, James Steel, and Benjamin Fuller. Cryptographic authentication from the iris. In International Conference on Information Security, pages 465{485. Springer, 2019.

[74] Karen P Hollingsworth, Kevin W Bowyer, and Patrick J Flynn. The best bits in an iris code. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(6):964{973, 2008.

[75] Ruud M Bolle, Sharath Pankanti, Jonathan H Connell, and Nalini K Ratha. Iris individuality: A partial iris model. In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., volume 2, pages 927{930. IEEE, 2004.

[76] Pawel Drozdowski, Christian Rathgeb, Antitza Dantcheva, Naser Damer, and Christoph Busch. Demographic bias in biometrics: A survey on an emerging challenge. IEEE Transactions on Technology and Society, 1(2):89{103, 2020.

[77] Sheikh Ziauddin and Matthew N Dailey. Iris recognition performance enhancement using weighted majority voting. In 2008 15th IEEE International Conference on Image Processing, pages 277{280. IEEE, 2008.

[78] Joseph Akinyele, Christina Garman, Ian Miers, Matthew Pagano, Michael Rushanan, Matthew Green, and Aviel Rubin. Charm: A framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 3, 06 2013.

[79] Hui Cui, Zhiguo Wan, Robert H. Deng, Guilin Wang, and Yingjiu Li. Ecient and expressive keyword search over encrypted data in cloud. IEEE Transactions on Dependable and Secure Computing, 15(3):409{422, 2018.

[80] Chunqiang Hu, Xiuzhen Cheng, Zhi Tian, Jiguo Yu, Kemal Akkaya, and Limin Sun. An attribute-based signcryption scheme to secure attribute-dened multicast communications. In Bhavani Thuraisingham, XiaoFeng Wang, and Vinod Yegneswaran, editors, Security and Privacy in Communication Networks, pages 418{437, Cham, 2015. Springer International Publishing.

[81] Fen Liu and Qi Wang. An identity-based batch verication scheme for vanets based on ring signature with ecient revocation. ArXiv, abs/2103.07653, 2021.

[82] Yutaka Kawai and Katsuyuki Takashima. Predicate- and attribute-hiding inner product encryption in a public key setting. In Zhenfu Cao and Fang-guo Zhang, editors, Pairing-Based Cryptography { Pairing 2013, pages 113{130, Cham, 2014. Springer International Publishing.

[83] Sohaib Ahmad, Chloe Cachet, Luke Demarest, Benjamin Fuller, and Ariel Hamlin. An implementation of proximity searchable encryption (PSE). https://github.com/chloecachet/pse, 2021.

[84] Kevin Lewi. FHIPE github. https://github.com/kevinlewi/fhipe, 2016.