Proximity Searchable Encryption for the Iris Biometric

Chloe Cachet
University of Connecticut
Storrs, CT, United States
chloe.cachet@uconn.edu

Sohaib Ahmad
University of Connecticut
Storrs, CT, United States
sohaib.ahmad@uconn.edu

Luke Demarest
University of Connecticut
Storrs, CT, United States
luke.h.demarest@gmail.com

Ariel Hamlin
Northeastern University
Boston, MA, United States
ahamlin@ccs.neu.edu

Benjamin Fuller
University of Connecticut
Storrs, CT, United States
benjamin.fuller@uconn.edu

ABSTRACT

Biometric databases collect people's information and allow users to perform proximity searches (finding 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 identifies and closes two gaps to 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 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.

ASIA CCS '22, May 30-June 3, 2022, Nagasaki, Japan

ACM ISBN 978-1-4503-9140-5/22/05…$15.00
https://doi.org/10.1145/3488932.3497754

CCS CONCEPTS

• **Information systems** → **Proximity search**; *Data encryption*; • **Security and privacy** → **Biometrics**; *Privacy-preserving protocols.*

KEYWORDS

Searchable encryption;biometrics;proximity search;inner product encryption

ACM Reference Format:
Chloe Cachet, Sohaib Ahmad, Luke Demarest, Ariel Hamlin, and Benjamin Fuller. 2022. Proximity Searchable Encryption for the Iris Biometric. In *Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security (ASIA CCS '22), May 30-June 3, 2022, Nagasaki, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3488932.3497754

1 INTRODUCTION

Biometrics are measurements of physical phenomena of the human body. We focus on the *iris* biometric in this work. 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 $\mathcal{D}$ 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 different biometrics are within distance t with low probability.

Learning stored biometric templates enables an attacker to reverse this value into a convincing biometric [1–3], enabling *presentation* attacks [4–6] that can compromise users' accounts and devices. Since biometrics cannot be updated, such a compromise lasts a lifetime.

Searchable encryption [7–10] enables servers to be queried without decrypting the data. For a distance metric $\mathcal{D}$, proximity searchable encryption returns all records that are within distance t. That is, for a dataset $x_1, ..., x_\ell$ for a query y, one should return all x_i such that $\mathcal{D}(x_i, y) \leq t$. 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] (fingerprints are usually compared by set difference, faces with L2 norm). Kim et al. proposed to use secret-key, function-hiding *inner product encryption* or $\mathsf{IPE}_{\mathsf{fh,sk}}$ for encrypted comparison of binary Hamming biometrics [11, 12]. $\mathsf{IPE}_{\mathsf{fh,sk}}$ allows computation of inner product without revealing underlying values. Inner product of vectors x, y in $\{-1, 1\}^n$ encodes Hamming distance:

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

More formally the functionality of $\mathsf{IPE}_{\mathsf{fh,sk}}$ is as follows: one generates $\mathsf{sk} \leftarrow \mathsf{Setup}(\cdot)$ and has two algorithms $\mathsf{ct}_x \leftarrow \mathsf{Encrypt}(x, \mathsf{sk})$ and $\mathsf{tk}_y \leftarrow \mathsf{TokGen}(y, \mathsf{sk})$ such that one can use Decrypt (without sk) to learn $\langle x, y \rangle$. That is, $\mathsf{Decrypt}(\mathsf{ct}_x, \mathsf{tk}_y) = \langle x, y \rangle$. One can use $\mathsf{IPE}_{\mathsf{fh,sk}}$ to build proximity search by encrypting $c_i \leftarrow \mathsf{Encrypt}(x_i, \mathsf{sk})$ and providing all c_i to the database server (additional data can be associated with x_i using traditional encryption). For queries y the client provides $\mathsf{tk}_y \leftarrow \mathsf{TokGen}(y, \mathsf{sk})$ 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.

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

1.1 Our Contribution

Multi Random Projection Inner Product Encryption. Daugman's seminal iris feature extractor [13, 14] produces a vector of length $n = 1024$, the open source OSIRIS [15] system uses $n = 32768$ by default, and recent neural network feature extractors [16] use $n = 2048$.

The most efficient $\mathsf{IPE}_{\mathsf{fh,sk}}$ schemes rely on *dual pairing vector spaces* [17] in bilinear groups. The secret key for such constructions is a random matrix $\mathbf{A} \in \mathbb{F}_q^{n \times n}$ and its inverse $\mathbf{A}^{-1}$; q is a large prime that is the order of the bilinear pairing. Setup for the scheme must invert a random $\mathbf{A} \in \mathbb{F}_q^{n \times n}$.

This operation is prohibitive for $n > 1000$, as is the case for iris feature extractors. For the most efficient known scheme which we call *Random Projection with Check* or RProjC [11], the authors' parallel implementation of key generation in FLINT [18] (on a modern 16 core machine), generating keys for $n = 240$, took roughly 3.5 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 3) that known techniques harm the quality of the biometric features, making the irises of different people appear similar. The false accept vs false reject rate tradeoff degrades, leaving the application with the choice of either not matching readings of the same iris or matching readings of difference individuals' irises. Both choices have consequences for the resulting application.

In Section 3.1 and Table 3, we show that for a small size dataset of 356 individuals using a feature extractor with $n = 64$, 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 4 introduces a new transform for inner product encryption that generates multiple matrices $\mathbf{A}_1, ..., \mathbf{A}_\sigma$ and their inverses during key generation where each $\mathbf{A}_i$ is an $(N+1) \times (N+1)$ matrix, where $N = \lceil n/\sigma \rceil$, instead of a single large pair $\mathbf{A}, \mathbf{A}^{-1}$. To hide partial information, both x and y are augmented when they are split into component vectors:

$$x'_i = 1 \,||\, x_{i*N}, ..., x_{i*N+(N-1)}$$
$$y'_i = \zeta_i \,||\, y_{i*N}, ..., y_{i*N+(N-1)}$$

for $i = 0, ..., \sigma - 1$ and $\zeta_0, ..., \zeta_{\sigma-1}$ is a linear secret sharing of 0 that is chosen in TokGen. The intuition is that any collection of $\sigma - 1$ or fewer components represents a random group element, so one cannot learn information about inner products between vector components. We show security of two prior IPE schemes with multi random projection (one in Section 4 and one in the full version of this work [19]).

We implemented two versions of proximity search building on this form of $\mathsf{IPE}_{\mathsf{fh,sk}}$. The first is a direct application of the RProjC [11] 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 [20, 21] which is a superset of the NIST Iris Evaluation Challenge [22]. 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, proximity search from $\mathsf{IPE}_{\mathsf{fh,sk}}$ for any searched value y, allows the server to compute the distance [11] 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 [25, 26]. While we are not aware of any leakage abuse attacks directly against proximity search, there are attacks against k-nearest neighbor databases [27, 28].[4] Distance allows one to easily compute the k-nearest points (with some error) so attacks that

[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.

[2] We have not evaluated sub-cubic matrix inversion in finite fields.

[3] Some prior work allows computation of approximate distance [23] using locality sensitive hashes [24], 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 [27–37].

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 7.

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 pairwise distance. A predicate IPE scheme produces ciphertexts ct_x and tokens tk_y which allow one to effectively check if $\langle x, y\rangle = 0$ (instead of revealing the inner product). Barbosa et al. [38] recently proposed such a scheme that is a modification of Kim et al.'s construction [11]. 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 $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ scheme. $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ 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', and create $y_i = y||i$. Then, $\langle x', y_i\rangle = (\langle x \,||\, \text{-1}, y \,||i\rangle = 0) \Leftrightarrow (\langle x, y\rangle = i)$. One can check all values in a set $\mathcal{I}$ by generating a token tk_{y_i} for each $i \in \mathcal{I}$. Setting $\mathcal{I} = \{n-2*0, ..., n-2*t\}$, yields a proximity check (these tokens are permuted before being sent to the server).

We call this construction *Multi Random Projection* or MRProj. The simplicity and generality of this construction is an advantage, it immediately benefits from efficiency improvements in inner product encryption and can be built from multiple computational assumptions. However, the size of tk_y and search time grow linearly with t. For the iris t is usually around $.3n$.

Since the server can see if the same tk_{y_i} matches different records, when two records are both within distance t, the server learns if they match the same distance (but not the specific distance). Thus, the resulting proximity scheme leaks two pieces of information:

Access Pattern [29, 30] 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 $\mathcal{D}(x_i, x_j) \leq 2t$. Preventing attacks that only require access pattern usually requires oblivious RAM [39] and its high storage and communication overhead.

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

No information is leaked about data that is not returned (beyond that it was not returned). Biometrics are well spread, 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 $n = 1024$ this corresponds 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 is not fast enough. In Section 8 we present avenues for improving search efficiency.

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

2 PRELIMINARIES

Let λ be the security parameter throughout the paper. We use $\mathtt{poly}(\lambda)$ and $\mathtt{negl}(\lambda)$ to denote unspecified functions that are polynomial and negligible in λ, respectively. For some $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \cdots, n\}$. Let $x \xleftarrow{\$} S$ denote sampling x uniformly at random from the finite set S. Let $q = q(\lambda) \in \mathbb{N}$ be a prime, then $\mathbb{G}_q$ denotes a cyclic group of order q. Let x denote a vector over $\mathbb{Z}_q$ such that $x = (x_1, \cdots, x_n) \in \mathbb{Z}_q^n$, the dimension of vectors should be apparent from context. Consider vectors $x = (x_1, \cdots, x_n)$ and $y = (y_1, \cdots, y_n)$, 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 defined as the distance between the bit vectors x and y of length n: $\mathcal{D}(x, y) = |\{i \,|\, x_i \neq y_i\}|$. We note that if a vector over $\{0, 1\}$ is encoded as $x_{\pm 1,i} = 1$ if $x_i = 1$ and $x_{\pm 1,i} = -1$ if $x_i = 0$ then it is true that $\langle x_{\pm 1}, y_{\pm 1}\rangle = n - 2\mathcal{D}(x, y)$.

Definition 1 (Asymmetric Bilinear Group). *Suppose $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are three groups (respectively) of prime order q with generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $g_T \in \mathbb{G}_T$ respectively. We denote a value x encoded in $\mathbb{G}_1$ with either g_1^x or $[x]_1$, we denote values encoded in $\mathbb{G}_2$ and $\mathbb{G}_T$ similarly. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a non-degenerate (i.e. $e(g_1, g_2) \neq 1$) bilinear pairing operation such that for all $x, y \in \mathbb{Z}_q$, $e([x]_1, [y]_2) = e(g_1, g_2)^{xy}$. We assume the group operations in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ and the pairing operation e are efficiently computable, then $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ defines an asymmetric bilinear group.*

Let $\mathcal{G}_{abg}$ be an algorithm that takes input 1^λ and outputs a description of an asymmetric bilinear groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ with security parameter λ.

2.1 Inner Product Encryption

Secret-key predicate encryption with function privacy supporting inner products queries was first proposed by Shen et al. [40]. This primitive allows one to check if the inner product between vectors is zero or not. 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.

Definition 2 (Secret key predicate encryption). *Let $\lambda \in \mathbb{N}$ be the security parameter, $\mathcal{M}$ be the set of attributes and $\mathcal{F}$ be a set of predicates. We define PE = (PE.Setup, PE.Encrypt, PE.TokGen, PE.Decrypt), a secret-key predicate encryption scheme, as follows:*
$\mathsf{PE.Setup}(1^\lambda) \to (sk, pp)$, $\mathsf{PE.Encrypt}(sk, x) \to ct_x$,
$\mathsf{PE.TokGen}(sk, f) \to tk_f$, and $\mathsf{PE.Decrypt}(pp, tk_f, ct_x) \to b$.

Scheme Name	Underlying IPE	IPE Type fh	sk	pred	Multi Random Proj Applied	Distance Hiding	Operation Time Setup	BuildIndex	Trapdoor	Search
RProjC	[11]	✓	✓	–	–	–	2×10^6	10.8	.07	235
MRProjC	[11]	✓	✓	–	✓	–	268	10.8	.08	241
MRProj	[38]	✓	✓	✓	✓	✓	268	10.8	22.4	12600

Table 1: Time (in seconds) for operations with $\ell = 356$ records stored at $n = 1024$. All algorithms are naturally parallelizable. Timing for the single base scheme is interpolated from smaller vector lengths. BuildIndex encrypts the dataset at initialization time, Trapdoor generates a search token, and Search finds 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. *Distance Hiding* indicates that the scheme does not reveal the distance between the stored value and the query.**

(1) Draws $\beta \xleftarrow{\$} \{0,1\}$,
(2) Computes $(\mathsf{sk}, \mathsf{pp}) \leftarrow \mathsf{PE.Setup}(1^\lambda)$, sends pp to $\mathcal{A}$,
(3) For $1 \le i \le s$, $\mathcal{A}$ chooses $x_i^{(0)}, x_i^{(1)} \in \mathcal{M}$,
(4) For $1 \le j \le r$, $\mathcal{A}$ chooses $f_j^{(0)}, f_j^{(1)} \in \mathcal{F}$,
(5) Denote $R := \left(x_1^{(0)}, x_1^{(1)}\right), \cdots, \left(x_r^{(0)}, x_r^{(1)}\right)$,
$S := \left(f_1^{(0)}, f_1^{(1)}\right), \cdots, \left(f_s^{(0)}, f_s^{(1)}\right)$.
(6) $\mathcal{A}$ sends R and S to C,
(7) $\mathcal{A}$ loses the game if R and S are not admissible,
(8) $\mathcal{A}$ receives

$$C^{(\beta)} := \{ct_i^{(\beta)} \leftarrow \mathsf{PE.Encrypt}\left(\mathsf{sk}, x_i^{(\beta)}\right)\}_{i=1}^{r},$$

$$T^{(\beta)} := \{tk_j^{(\beta)} \leftarrow \mathsf{PE.TokGen}\left(\mathsf{sk}, f_j^{(\beta)}\right)\}_{j=1}^{s}.$$

(9) $\mathcal{A}$ returns $\beta' \in \{0,1\}$,
(10) Her *advantage* is

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Exp}_{IND}^{\mathsf{PE}}}(\lambda) = \Big| \Pr[\mathcal{A}(1^\lambda, T^{(0)}, C^{(0)}) = 1] - \Pr[\mathcal{A}(1^\lambda, T^{(1)}, C^{(1)}) = 1] \Big|$$

Figure 1: Definition of $\mathbf{Exp}_{IND}^{\mathbf{PE}}$ for predicate encryption.

We require the scheme to have the following properties:

Correctness: *For any $x \in \mathcal{M}, f \in \mathcal{F}$,*

$$\Pr\left[f(x) = b \;\middle|\; \begin{matrix} ct_x \leftarrow PE.Encrypt(sk,x) \\ tk_f \leftarrow PE.TokGen(sk,f) \\ b \leftarrow PE.Decrypt(pp, tk_f, ct_x) \end{matrix}\right] \ge 1 - negl(\lambda).$$

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

$$PE.Decrypt(pp, tk_j^{(0)}, ct_i^{(0)}) = PE.Decrypt(pp, tk_j^{(1)}, ct_i^{(1)}).$$

The above definition is called *full security* in the language of Shen, Shi, and Waters [40]. Note that this definition is selective (not adaptive), as the adversary specifies two sets of plaintexts and functions apriori. The relevant primitive for us is $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ which uses the above definition restricted to the class of predicates $\mathcal{F} = \{f_y \mid y \in \mathbb{Z}_q^n\}$ such that for all vectors $x \in \mathbb{Z}_q^n$, $f_y(x) = 1$ when $\langle x, y \rangle = 0$, $f_{y,t}(x) = 0$ otherwise.
We use (IPE.Setup, IPE.Encrypt, IPE.TokGen, IPE.Decrypt) to refer to the corresponding tuple of algorithms.

2.2 Proximity searchable encryption

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

Definition 3 (History). *Let $X \in \mathcal{M}$ be a list of keywords drawn from space $\mathcal{M}$, let $\mathcal{F}$ be a class of predicates over $\mathcal{M}$. An m-query history over $\mathcal{W}$ is a tuple History $= (X, F)$, with $F = (f_1, \cdots, f_m)$ a list of m predicates, $f_i \in \mathcal{F}$.*

Definition 4 (Access pattern). *Let $X \in \mathcal{M}$ be a list of keywords. The access pattern induced by an m-query history History $= (X, F)$ is the tuple AccPatt(History) $= (f_1(X), \cdots, f_m(X))$*

Definition 5 (Distance Equality). *Let History$^{(0)}$, History$^{(1)}$ be m-query histories for predicates of the type $f_{y,t}(x) = (\mathcal{D}(x,y) \overset{?}{\le} t)$. Let, DisEq(History$^{(0)}$, History$^{(1)}$) $= 1$ if and only if for each j it is true that*

$$\left\{(i,k) \;\middle|\; \begin{matrix} (\mathcal{D}(x_i^{(0)}, y_j^{(0)}) = \mathcal{D}(x_k^{(0)}, y_j^{(0)}) \wedge \mathcal{D}(x_i^{(1)}, y_j^{(1)}) \ne \mathcal{D}(x_k^{(1)}, y_j^{(1)})) \\ \vee \\ (\mathcal{D}(x_i^{(0)}, y_j^{(0)}) \ne \mathcal{D}(x_k^{(0)}, y_j^{(0)}) \wedge \mathcal{D}(x_i^{(1)}, y_j^{(1)}) = \mathcal{D}(x_k^{(1)}, y_j^{(1)})) \end{matrix}\right\},$$

is the empty set.

Definition 6 (Proximity Searchable Encryption). *Let*

- *$\lambda \in \mathbb{N}$ be the security parameter,*
- *$\mathcal{DB} = (M_1, \cdots, M_\ell)$ be a database of size ℓ,*
- *Keywords $X = (x_1, \cdots, x_\ell)$, such that $x_i \in \mathbb{Z}_q^n$ relates to M_i,*
- *$\mathcal{F} = \{f_{y,t} \mid y \in \mathbb{Z}_q^n,\ t \in N\}$ be a family of predicates such that, for a keyword $x \in \mathbb{Z}_q^n$, $f_{y,t}(x) = 1$ if $\mathcal{D}(x,y) \le t$, 0 otherwise.*

The algorithms PSE = (PSE.Setup, PSE.BuildIndex, PSE.Trapdoor, PSE.Search) defines a proximity searchable encryption scheme: PSE.Setup$(1^\lambda) \to (sk, pp)$, PSE.BuildIndex$(sk, X) \to I_X$, PSE.Trapdoor$(sk, f_{y,t}) \to tk_{y,t}$, and PSE.Search$(pp, Q_{y,t}, I_X) \to J_{X,y,t}$. We require the scheme to have the following properties:

Correctness *Define $J_{X,y,t} = \{i | f_{y,t}(x_i) = 1, x_i \in X\}$. PSE is correct if for all X and $f_{y,t} \in \mathcal{F}$:*

$$\Pr\left[J' = J_{X,y,t} \;\middle|\; \begin{matrix} I_X \leftarrow PSE.BuildIndex(sk,X) \\ Q_{y,t} \leftarrow PSE.Trapdoor(sk, f_{y,t}) \\ J' \leftarrow PSE.Search(pp, Q_{y,t}, I_X) \end{matrix}\right] \ge 1 - negl(\lambda).$$

Security for Admissible Queries *Any PPT adversary $\mathcal{A}$ has only $negl(\lambda)$ advantage in the experiment Exp_{IND}^{PSE} defined in Figure 2, for $\ell = \mathtt{poly}(\lambda)$ and $m = \mathtt{poly}(\lambda)$.*

(1) *Draws* $\beta \xleftarrow{\$} \{0,1\}$,
(2) *Computes* $(sk, pp) \leftarrow PSE.Setup(1^\lambda)$ *and sends pp to* $\mathcal{A}$.
(3) $\mathcal{A}$ *chooses and outputs* $History^{(0)}, History^{(1)}$.
(4) $\mathcal{A}$ *loses the game if*

$$AccPatt(History^{(0)}) \neq AccPatt(History^{(1)}) \vee DisEq(History^{(0)}, History^{(1)}) = 0$$

(5) $\mathcal{A}$ *receives* $I^{(\beta)}$ *and* $Q^{(\beta)}$.
(6) $\mathcal{A}$ *outputs* $\beta' \in \{0,1\}$
(7) *Her advantage in the game is:*

$$\mathrm{Adv}^{Exp^{PSE}_{IND}}_{\mathcal{A}}(\lambda) = \left| \Pr[\mathcal{A}(1^\lambda, I^{(0)}, Q^{(0)}) = 1] - \Pr[\mathcal{A}(1^\lambda, I^{(1)}, Q^{(1)}) = 1] \right|$$

Figure 2: Definition of $\mathbf{Exp}^{\mathbf{PSE}}_{IND}$.

3 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 [13, 14] 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 [14, 41]. Traditional iris recognition consists of three phases:

Segmentation takes the image and identifies 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 fixed size rectangular array. This can roughly be thought of as unrolling the iris.

Feature Extraction transforms the rectangular array into a fixed 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 identification systems the tradeoff is between FRR and FAR. FRR is how frequently readings of the *same* biometric are regarded as different. FAR is how frequently readings of *different* 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 $\{x_i | \mathcal{D}(x_i, y) \leq t\}$ for some metric $\mathcal{D}$ 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 tradeoff we introduce the feature extractor and dataset used in this analysis.

Feature Extractor For the feature extractor, we use the recent pipeline called ThirdEye [16, 42], which is publicly available [43]. The software produces a 1024 dimensional real valued feature vector. We convert this to a binary vector by setting $f'_i = 1$ if $f_i > \mathrm{Exp}[f_i]$ where the $\mathrm{Exp}[f_i]$ is the expectation of the individual feature, otherwise $f'_i = 0$. We train the feature extractor as specified in [16].

Biometric Database There are many iris datasets collected across a variety of conditions. In this work we use the NotreDame 0405 dataset [20, 21] which is a superset of the NIST Iris Evaluation Challenge [22]. This dataset consists of images from 356 biometrics (we consider left and right eyes as separate biometrics) with 64964 images in total. (See Appendix B for similar results with the IITD dataset [44].) Figure 3(a) shows the histograms for the testing portions of the feature extractor outputs. The blue histogram contains comparisons between different readings of the same biometric while the red histogram contains comparisons between different biometrics. Let $t' = t/1024$ be the fractional Hamming distance, the FRR is the fraction of the blue histogram to the right of t' and the FAR is the fraction of the red histogram to the left of t'. There is overlap between the red and blue histogram indicating that there is a tradeoff between FRR and FAR.

3.1 Performance of Biometric Identification with Small Dimension

The efficiency of IPE based proximity search critically depends on the number of features n (see Table 4). In our experiments we estimate Setup for $n = 1024$ for the schemes of Kim et al. [11] and Barbosa et al. [38] to take 23 days on a modern server machine (see details in Section 6). 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 motivates our approach to reduce the complexity of Setup in Section 4. Our analysis consists of two major parts:

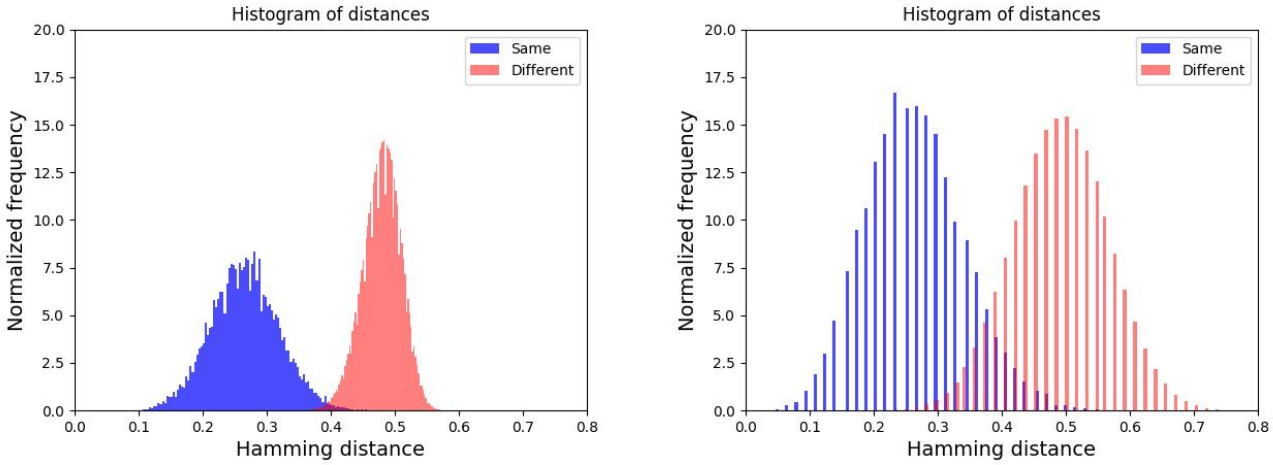
(1) We compare different mechanisms for reducing the size of feature vectors using $n = 64$ as the target dimension.
(2) Using the best feature reduction mechanism we compare the FRR/FAR tradeoff for $n < 1024$, showing direct impacts for the correctness and security of the resulting biometric search.

3.1.1 Dimensionality Reduction Method. We consider four different mechanisms for dimension reduction and consider their impact on FRR/FAR. For all techniques, the most important phenomena is that variance of Different comparisons increases as the sample size decreases.[5] Compare Figure 3(a) and Figure 3(b). This makes the tails of Same and Different wider leading to worse identification. The four mechanisms we consider are:[6]

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. [46] and Bolle et al. [47] propose the concept of "fragile bits" which are more likely to be susceptible to bit flips. Their work is based on the Gabor based feature extractor (described at the beginning of

[5] 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 [45].
[6] For all experiments we computed the mechanism four times and report the average in Table 2.



(a) ND 0405 Histogram with $n = 1024$

(b) ND 0405 Histogram with $n = 64$

Figure 3: Hamming distance distribution for images from the same iris in blue, and different irises in red. Histograms are produced using ThirdEye [16]. Resulting histograms for the ND 0405 dataset. Figure 3(a) shows the histogram when $n = 1024$ with a small overlap between distances comparisons of the same iris and different irises. This overlaps is increased substantially when $n = 64$ in in Figure 3b). Figure 3b) is produced using the E method.

this section) while ThirdEye [16] is a convolutional neural network.

We select the 64 bits which have the least probability of flipping. 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 different 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. That is, these are bits where the difference between intra class and inter class error is the highest. That is, we select the bits that maximize the following difference:

$$\max_i \left(\Pr_{x,y \leftarrow \text{Different}}[x_i \neq y_i] - \Pr_{x,y \leftarrow \text{Same}}[x_i \neq y_i] \right)$$

The intuition is that bits are the most useful as they maximize the difference in probability of error between the same and different comparisons. The hope is to overcome the weakness of the prior approach which did not consider the entropy of bits across different 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 [16] 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 (for train) to denote this technique.

FRR	False Accept Rate										
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: FRR for different output sizes and probabilities of leakage for the ND0405 datasets. Summary of false reject rates for queries drawn from *Same* distribution. We vary a threshold t, report the false reject rate (FRR) when allowing for the corresponding FAR. The original $n = 1024$ system is presented for comparison.

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 different vector sizes.

3.1.2 Impact of reducing n. We now show that decreasing n using the E method hurts the identification quality of the iris biometric. First we note that an FRR of $\leq .10$ requires a distance tolerance of $t \geq .3n$ (see the histograms in Figure 3). However, comparisons between different irises are tightly centered around $t = .5n$. This means for a dataset $\{x_i\}_{i=1}^{\ell}$ for most pairs x_i, x_j there exists some value x^* such that $\mathcal{D}(x_i, x^*) \leq t$ and $\mathcal{D}(x_j, x^*) \leq t$. This means for most pairs x_i, x_j, there is some query that will cause them both to be returned.

	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
σ^2	.75	.74	.42	.23	.17	.083	.076	.019

Table 3: Effect 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 goal of this subsection is to understand behavior on actual queries. We consider a distribution over x^* of different readings of individuals stored in 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 affects 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 consider correctness of the system at different feature vector lengths n. We select a random reading of each biometric to represent the *encrypted dataset*. We first select a t that yields at most $\leq 10\%$ 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 $\mathcal{I} \subset \{1, ..., 356\}$ of size 150 randomly.
(3) For each i in $\mathcal{I}$:
   (a) Select 3 random readings of iris i, denoted x_i^* (removing reading that is encrypted):[7]
   (b) For all j if $\mathcal{D}(x_i^*, x_j) \leq t$ and $\mathcal{D}(x_i^*, x_i) \leq t$ $C_{i,j} = C_{i,j} + 1$.
(4) Compute $\texttt{ACount} = \sum_{i=0}^{355} \left(\sum_{j=0, j\geq i}^{355} C_{i,j}\right) / (3 * 150)$.

The value ACount represents how frequently a record of a different biometric would be returned by an in use search system. For both correctness and security considers one desires ACount to be as close to 0 as possible. 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 $x_{i,1}, ..., x_{i,k}$ are associated with a single record. Until now, we've implicitly assumed that the database has only one reading of a biometric. We now briefly consider the implications of leakage between readings of the same biometric. That is, $x_{i,1}, ..., x_{i,k}$ are readings from the same biometric and associated with a record r_i in the biometric database. First note that $x_{i,\alpha}$ and $x_{i,\beta}$ are likely to be close together (because readings of the same biometric are similar).

One may able to infer information about $x_{i,1}, ..., x_{i,k}$ from access pattern and distance equality leakage. One may be able to learn the relative positioning of the different readings by which values $\mathcal{I}$ 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 $x_{i,1}, ..., x_{i,k}$. Both of these leakage profiles allow an adversary to construct geometry of a biometric's different 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 sensitive attributes about the corresponding person. Biometric systems frequently demonstrate systemic bias [48]. As one example most datasets draw from volunteer undergraduates students. Systems accuracy varies based on sensitive attributes such as gender, race, and age (see [48, Table 1]). Thus one may be able to infer sensitive attributes based on the relative size of $|\mathcal{I}|/k$.

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 [49] and make other values associated data that are not searchable.

4 MULTI RANDOM PROJECTION IPE

As described in the Introduction, we show a general technique improving Setup efficiency for IPE schemes where ciphertexts and tokens are projected into dual vector spaces by a pair of matrices $\mathbf{A}, \mathbf{A}^{-1}$. 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. [38, Section 4].[8]

Construction The construction is in Figure 4. We first argue correctness and then security. For security we show the scheme satisfies a stronger simulation based definition of security, as in the work of Barbosa et al. [38].

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

$$\begin{aligned}\Pi_{\ell=1}^{\sigma}\Pi_{i=1}^{N} e(\mathsf{tk}_\ell[i], \mathsf{ct}_\ell[i]) &= g_T^{\sum_{\ell=1}^{\sigma} \beta\cdot(x'_\ell)^T\cdot\mathbb{B}_\ell^*\cdot\mathbb{B}_\ell^T\cdot\alpha\cdot(y'_\ell)} \\ &= g_T^{\sum_{\ell=1}^{\sigma} \beta\cdot(x'_\ell)^T\cdot\alpha\cdot(y'_\ell)} = g_T^{\alpha\beta\sum_{\ell=1}^{\sigma}\zeta_\ell + \langle x_\ell, y_\ell\rangle} \\ &= g_T^{\alpha\beta\cdot\langle x,y\rangle + \alpha\beta\cdot\sum_{\ell=1}^{\sigma}\zeta_\ell} = g_T^{\alpha\beta\cdot\langle x,y\rangle}\end{aligned}$$

If $\langle x, y\rangle = 0$ then $\Pi_{\ell=1}^{\sigma}\Pi_{i=1}^{N} e(\mathsf{tk}_\ell[i], \mathsf{ct}_\ell[i]) = e(g_1, g_2)^0 = 1$, which is the identity element in $\mathbb{G}_T$ and is easily detectible and $\top \leftarrow \mathsf{Decrypt}(\mathsf{pp}, \mathsf{tk}, \mathsf{ct})$ with probability 1. If $\langle x, y\rangle \neq 0$, then the probability that $\top \leftarrow \mathsf{Decrypt}(\mathsf{pp}, \mathsf{tk}, \mathsf{ct})$ is $\Pr[\alpha\beta\cdot\langle x, y\rangle = 0] \leq \frac{2}{q}$.

Definition 7 (Simulation-based security). *Let IPE = (IPE.Setup, IPE.TokGen, IPE.Encrypt, IPE.Decrypt) be a predicate IPE scheme over $\mathbb{Z}_q^n$. Then IPE is SIM-secure if for all PPT adversaries $\mathcal{A}$, there exist a simulator $\mathcal{S}$ such that for the experiment Exp_{SIM}^{IPE} described in figure 5, the advantage of $\mathcal{A}$ ($\mathsf{adv}_{\mathcal{A}}^{Exp_{SIM}^{IPE}}$) is*

$$\left|\ \Pr[1 \leftarrow \mathsf{Real}_{IPE,\mathcal{A}}(1^\lambda)] - \Pr[1 \leftarrow \mathsf{Ideal}_{IPE,\mathcal{A}}(1^\lambda)]\ \right| \leq negl(\lambda).$$

[7] 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.

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

Setup$(1^\lambda, n, \sigma)$:

(1) Sample $(G_1, G_2, G_T, q, e) \leftarrow \mathcal{G}_{abg}$ and randomly sample generators $g_1 \in G_1$ and $g_2 \in G_2$.
(2) For $1 \le \ell \le \sigma$, randomly samples an invertible square matrix $\mathbb{B}_\ell \in \mathbb{Z}_q^{N\times N}$ and sets $\mathbb{B}_\ell^* = (\mathbb{B}_\ell^{-1})^T$, with $N = \lceil n/\sigma \rceil + 1$.
(3) Outputs $\mathsf{pp} = (G_1, G_2, G_T, q, e, n, \sigma)$ as public parameters and $\mathsf{sk} = (g_1, g_2, \{\mathbb{B}_\ell, \mathbb{B}_\ell^*\}_{\ell=1}^{\sigma})$.

TokGen(pp, sk, y):

(1) Sample $\alpha \xleftarrow{\$} \mathbb{Z}_q$.
(2) Splits y into σ subvectors y_ℓ of size $\lceil n/\sigma \rceil$ and pads with zeroes if needed.
(3) For $1 \le \ell \le \sigma$, defines $y'_\ell = 1 \,||\, y_\ell$ and sets $\mathsf{tk}_\ell = [\ \alpha \cdot (y'_\ell)^T \cdot \mathbb{B}_\ell\ ]_1$, a vector in G_1.
(4) Outputs $\mathsf{tk} = (\mathsf{tk}_1, \cdots, \mathsf{tk}_\sigma)$.

Encrypt(pp, sk, x):

(1) Samples $\beta \xleftarrow{\$} \mathbb{Z}_q$.
(2) Splits x into σ subvectors x_ℓ of size $\lceil n/\sigma \rceil$, and pads with zeroes if needed.
(3) For $1 \le \ell \le \sigma - 1$, samples $\zeta_\ell \xleftarrow{\$} \mathbb{Z}_q$ then sets $\zeta_\sigma = -\sum_{\ell=1}^{\sigma-1} \zeta_\ell$.
(4) For $1 \le \ell \le \sigma$ defines $x'_\ell = \zeta_\ell \,||\, x_\ell$ and sets $\mathsf{ct}_\ell = [\ \beta \cdot (x'_\ell)^T \cdot \mathbb{B}_\ell^*\ ]_2$, a vector in G_2.
(5) Outputs $\mathsf{ct} = (\mathsf{ct}_1, \cdots, \mathsf{ct}_\sigma)$.

Decrypt(pp, tk, ct):

Computes $\left(\Pi_{\ell=1}^{\sigma} \Pi_{i=1}^{N} e(\mathsf{tk}_\ell[i], \mathsf{ct}_\ell[i])\right)$ and returns $\top$ if the results is equal to $1 \in \mathbb{G}_T$, $\perp$ otherwise.

Figure 4: Construction of MRProj.

$\mathsf{Real}_{\mathsf{IPE},\mathcal{A}}(1^\lambda)$
$(\mathsf{sk}, \mathsf{pp}) \leftarrow \mathsf{IPE.Setup}(1^\lambda)$
$b \leftarrow \mathcal{A}^{\mathsf{IPE.TokGen}(\mathsf{sk},\cdot),\mathsf{IPE.Encrypt}(\mathsf{sk},\cdot)}(1^\lambda)$
Output b
$\mathsf{Ideal}_{\mathsf{IPE},\mathcal{A}}(1^\lambda)$
$(\mathsf{sk}, \mathsf{pp}) \leftarrow \mathsf{IPE.Setup}(1^\lambda)$
$b \leftarrow \mathcal{A}^{\mathcal{S}(\Phi(\cdot))}(1^\lambda)$
Output b

Figure 5: Definition of experiment $\mathbf{Exp}_{SIM}^{\mathbf{IPE}}$. Φ denotes the information leakage received by the simulator $\mathcal{S}$ such that $\Phi(i, j) = f_{y_j}(x_i)$ for all i, j.

Kim et. al. [12, Remark 2.5] show that Definition 7 implies Definition 2 so we argue that the scheme in Figure 4 satisfies Definition 7.

Theorem 1. *In the Generic Group Model for asymmetric bilinear groups the construction in Figure 4 is a secure $\mathsf{IPE}_{fh,sk,pred}$ scheme according to Definition 7 for the family of predicates $\mathcal{F} = \{f_y | y \in \mathbb{Z}_q^n\}$ such that for all vectors $x \in \mathbb{Z}_q^n$, $f_y(x) = (\langle x, y\rangle \stackrel{?}{=} 0)$.*

The proof of Theorem 1 is deferred until Appendix C.

5 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 [40], or $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$, allows one to test if the inner product between two vectors is equal to zero. By appending a value to the first 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 $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ to construct PSE for Hamming distance[9]. 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 $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$. Keywords are similarly encoded but this time a distance from the range is appended to them, and tokens generated as part of the underlying $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ 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 $\mathcal{F}$ 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.BuildIndex, PSE.Trapdoor, PSE.Search), a secure proximity searchable encryption scheme for the Hamming distance, such that for any PPT adversary $\mathcal{A}_{PSE}$ for Exp_{IND}^{PSE}, there exists a PPT adversary $\mathcal{A}_{IPE}$ for Exp_{IND}^{IPE}, such that for any security parameter $\lambda \in \mathbb{N}$,*

$$\mathsf{Adv}_{\mathcal{A}_{PSE}}^{Exp_{IND}^{PSE}} = \mathsf{Adv}_{\mathcal{A}_{IPE}}^{Exp_{IND}^{IPE}}$$

The proof of Theorem 2 is deferred to Appendix D. Table 4 presents the resulting efficiency of distance hiding PSE schemes based on different $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ constructions. This table corresponds to $t + 1$ tokens with all operations on dimension $n + 1$.

6 IMPLEMENTATION

This section presents an implementation and an evaluation of the PSE scheme proposed in this paper. We implemented the MRProj

[9] 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 identifiable by the server.

PSE.Setup$(1^\lambda) \rightarrow$ (sk, pp):

Run and output (sk, pp) $\leftarrow$ IPE.Setup(1^λ).

PSE.Trapdoor(sk, $f_{y,t}$) $\rightarrow Q_{y,t}$:

(1) For $0 \le j \le t$ compute $d_j = n - 2j$,
(2) Set $D = (d_0, ..., d_t)$,
(3) Sample random permutation $\pi : [0, t] \rightarrow [0, t]$,
(4) Compute $D^* = \pi(D) = \{d_0^*, \cdots, d_t^*\}$,
(5) Encode y as $y^* \in \{-1, 1\}^n$,
(6) For $0 \le j \le t$ call
$\mathsf{tk}_j \leftarrow$ IPE.TokGen(sk, $y^* || d_j^*$),
(7) Output $Q_{y,t} = (\mathsf{tk}_0, \cdots, \mathsf{tk}_t)$.

PSE.BuildIndex(sk, X) $\rightarrow I_X$:

(1) For each keyword $x_i \in X$, $i \in \{1, \cdots, \ell\}$,
encode $x_i^* \in \{-1, 1\}^n$,
compute $\mathsf{ct}_i \leftarrow$ IPE.Encrypt(sk, $x_i^* ||$ -1).
(2) Outputs $I_X = (\mathsf{ct}_1, \cdots, \mathsf{ct}_\ell)$.

PSE.Search(pp, $Q_{y,t}, I_X$) $\rightarrow J_{X,y,t}$:

(1) Initialize $J_{X,y,t} = \emptyset$.
(2) For each $\mathsf{ct}_i \in I_X$ and for each $\mathsf{tk}_j \in Q_{y,t}$,
call $b_j \leftarrow$ IPE.Decrypt(pp, $\mathsf{tk}_j, \mathsf{ct}_i$).
If $b_j = 1$, add i to $J_{X,y,t}$, continue to ct_{i+1}.
(3) Outputs $J_{X,y,t}$.

Figure 6: Construction of proximity search from $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$.

	Underlying IPE scheme				
	MRProj	RProj [38, Section 4]	[38, Section 5]	[50]	[40]
group order	Prime	Prime	Prime	Prime	Composite
Setup	$\sigma((n+1)/\sigma)^3$	$(n+1)^3$	$(n+1)^3$	$(6n+6)^3$	$4n+8$
BuildIndex	$\ell(n+\sigma+1)$	$\ell(n+1)$	$\ell(12n+21)$	$6\ell(n+1)$	$\ell(32n+36)$
Trapdoor	$(t+1)(n+\sigma+1)$	$(t+1)(n+1)$	$(t+1)(12n+21)$	$6(t+1)(n+1)$	$(t+1)(24n+40)$
Search	$\ell(t+1)(n+\sigma+1)$	$\ell(t+1)(n+1)$	$\ell(t+1)(6n+12)$	$6\ell(t+1)(n+1)$	$\ell(t+1)(4n+8)$
$\lvert\mathsf{sk}\rvert$	$2(n+1)^2/\sigma + 4n + 2\sigma + 6$	$2(n+1)^2 + 2$	$24n+42$	$60(n+1)^2$	$4n+8$
$\lvert\mathcal{I}\rvert$	$\ell(n+\sigma+1)$	$\ell(n+1)$	$\ell(6n+12)$	$6\ell(n+1)$	$\ell(2n+4)$
$\lvert\mathsf{tk}_{y,t}\rvert$	$(t+1)(n+\sigma+1)$	$(t+1)(n+1)$	$(t+1)(6n+12)$	$6(t+1)(n+1)$	$(t+1)(2n+4)$

Table 4: PSE scheme efficiency for keywords of size n depending on underlying $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ 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 scheme of Shen, Shi, and Waters [40] uses a composite order group whose order is the product of four large primes. The number n is the length of the biometric template, σ is the number of bases in the multi random projection scheme, t is the desired distance tolerance, and ℓ is the total number of records in the database.

construction described in section 4 and a PSE (see section 5) scheme using it in Python 3. These implementations can be found in a Github repository [51]. Our IPE implementations uses the Charm [52] and FLINT [18] libraries for the pairing group operations and finite field arithmetic in $\mathbb{Z}_q$. For comparison purposes, we used the pairing group over the asymmetric curve MNT159, the same as in Kim et al.'s FHIPE implementation [53].

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 efficiency for different 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. To be as realistic as possible, we used iris readings from the ND 0405 as input vectors to the benchmarking tests.

6.1 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 3).

Timing. We evaluate the timing efficiency of our PSE construction with and without the multi random projection technique. Table 5 reports the timings for all four algorithms of the PSE scheme. MRProj corresponds to the PSE construction presented in this paper. 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 σ values that yield sub-vectors lengths of approximately 40. We make three main observations.

(1) Setup and BuildIndex have comparable performance for MRProj and MRProjC (the only difference is adding 1 to underlying dimension). However, Trapdoor 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 Search takes 4 minutes, MRProj Search takes 3.5 hours. Both approaches scans the whole database which is

			Time							Sizes		
			MRProj				MRProjC		RProjC [11]	MRProj		RProjC [11]
n	σ	t	Setup	BuildIndex	Trapdoor	Search	Trapdoor	Search	Setup	|EncDB|	|sk|	|sk|
128	3	38	75	1.5	.36	234	.01	31	4×10^3	5.9 MB	560 KB	1.6 MB
192	5	57	47	2.2	.8	495	.01	46	1.3×10^4	8.9 MB	770 KB	3.6 MB
256	7	76	57	2.9	1.4	850	.02	62	3.2×10^4	12 MB	980 KB	6.4 MB
384	10	115	94	4.4	3.1	1870	.03	92	1.1×10^5	18 MB	1.5 MB	14 MB
512	13	153	153	5.7	5.7	3282	.04	140	2.6×10^5	24 MB	2.1 MB	26 MB
768	19	230	269	8.6	13.4	7210	.06	185	8.6×10^5	36 MB	3.2 MB	57 MB
1024	25	307	268	10.8	22.4	12600	.08	241	2.0×10^6	47 MB	4.3 MB	100 MB

Table 5: Operations timing (in seconds) and sizes (in Megabytes/Kilobytes) for different vector sizes. n is the vector length, σ the number of bases used, and $t = .30$ the distance tolerance. Setup and BuildIndex procedures for MRProj and MRProjC schemes are the same procedures, MRProjC uses vectors whose length is 1 fewer. We only report these algorithms for MRProj. Timing and storage for the MRProjC Setup is interpolated. Measured $n = 10$ to 240 in steps of 10. For timing, cubic fit with coefficients $y = .003x^3 - .578x^2 + 36x - 557$ with $R^2 = .996$. For storage, quadratic fit with coefficients $y = 96x^2 + 192x + 573$ with $R^2 = 1$.

problematic for large datasets. We discuss possible solutions in Section 8.

(3) 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 five minutes for input vectors 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 efficiency 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 efficient to perform multiple inversions of small matrices than a single inversion of a bigger one.

Storage. We evaluate the impact of the multi random projection PSE construction on storage efficiency. As can be seen on table 5, the impact is low for small input vectors, however, it makes a big difference for larger ones. Indeed, when the size of the Barbosa key (key generated without the multi random projection technique) grows quadratically with the vector size, the size of the key generated with the multi random projection technique grows with $(n/\sigma)^2 * \sigma \approx n^2/\sigma$. For vectors of size 1024, we consider $\sigma = 25$ and the secret key generated with the multi random projection technique is 23.2 times smaller than the single basis key, confirming the asymptotic analysis.

7 FURTHER RELATED WORK

In this section we review further related work on proximity search. We defer discussion of leakage abuse attacks to Appendix A. Li et al. [54], Wang et al. [55] and Boldyreva and Chenette [56] 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 $\{x_j \mid \mathcal{D}(x_i, x_j) \leq t\}$ are added as keywords associated to x_i.
(2) The second approach requires searchable encryption supporting disjunctive search. It inserts just x_i, but when searching for y it searches for the disjunction $\vee_{x_i \mid \mathcal{D}(x_i, y) \leq t}\ x_i$.

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 $\vee_{x_i \mid \mathcal{D}(x_i, y) \leq t}\{x_i\}$ usually grows exponentially in t. In existing disjunctive schemes, the size of the query grows with the size of the disjunction [10], making this approach only viable for constant values of t.

Kuzu et al.'s [23] solution relies on *locality sensitive hashes* [24]. A locality sensitive hash ensures that close values have a higher probability to produce collisions than values that are far apart. Thus, a scheme 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 [57] propose a variant of inner product encryption that reveals if the distance is less than t only. However, their security is based on $\mathbf{A}x_i$ and $y\mathbf{B}$ hiding x_i and y for secret square $\mathbf{A}$ and $\mathbf{B}$. Security is heuristic with no underlying assumption or proof of information theoretic security.

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 specific 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 3). This means that with a fixed 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 [11], 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. [11], additionally the distance is leaked between queries and all points in the database. Based on prior work on trilaterilation, 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 offer 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 fixed 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 profile for our MRProj 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 significant 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 specific distance) and so we see a significant (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 [27, 28]) and interactive solutions where the client can guide the search. In parallel work, Boldyreva and Ting [58] proposed such a scheme that hides all leakage using oblivious data structures in conjunction with locality sensitive hashes [24].

Acknowledgements. The authors thank the reviewers for their helpful comments and suggestions. The authors wish to thank Daniel Wichs for important preliminary discussion. L.D. is supported by the Harriott Fellowship. C.C. is supported by NSF Award #1849904 and a fellowship from Synchrony Inc. S.A. is supported by a fellowship from Synchrony Inc. and the State of Connecticut. B.F. is supported by NSF Award #1849904 and ONR Grant N00014-19-1-2327. A.H. is supported by NSF Award #1750795.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19020700008. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government.

The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

REFERENCES

[1] J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia, "From the iriscode to the iris: A new vulnerability of iris recognition systems," *Black Hat Briefings USA*, vol. 1, 2012.
[2] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the reconstruction of face images from deep face templates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 5, pp. 1188–1202, 2018.
[3] S. Ahmad and B. Fuller, "Resist: Reconstruction of irises from templates," in *2020 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2020, pp. 1–10.
[4] S. Venugopalan and M. Savvides, "How to generate spoofed irises from an iris code template," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 385–395, 2011.
[5] Y. Huang, A. K. Wai-Kin, and K.-Y. Lam, "From the perspective of CNN to adversarial iris images," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018, pp. 1–10.
[6] S. Soleymani, A. Dabouei, J. Dawson, and N. M. Nasrabadi, "Adversarial examples to fool iris recognition systems," in *2019 International Conference on Biometrics (ICB)*. IEEE, 2019, pp. 1–8.
[7] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*. IEEE, 2000, pp. 44–55.
[8] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, pp. 895–934, 01 2011.
[9] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, pp. 1–51, 2014.
[10] B. Fuller, M. Varia, A. Yerukhimovich, E. Shen, A. Hamlin, V. Gadepally, R. Shay, J. D. Mitchell, and R. K. Cunningham, "SoK: Cryptographically protected database search," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 172–191.
[11] S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu, "Function-hiding inner product encryption is practical," in *International Conference on Security and Cryptography for Networks*. Springer, 2018, pp. 544–562.
[12] S. Kim, K. Lewi, A. Mandal, H. W. Montgomery, A. Roy, and D. J. Wu, "Function-hiding inner product encryption is practical." *IACR Cryptology ePrint Archive*, vol. 2016, p. 440, 2016.
[13] J. Daugman, "Results from 200 billion iris cross-comparisons," 01 2005.
[14] ——, "How iris recognition works," in *The essential guide to image processing*. Elsevier, 2009, pp. 715–739.
[15] N. Othman, B. Dorizzi, and S. Garcia-Salicetti, "Osiris: An open source iris recognition software," *Pattern Recognition Letters*, vol. 82, pp. 124–131, 2016.
[16] S. Ahmad and B. Fuller, "Thirdeye: Triplet-based iris recognition without normalization," in *IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2019.
[17] T. Okamoto and K. Takashima, "Dual pairing vector spaces and their applications," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 98, no. 1, pp. 3–15, 2015.
[18] W. B. Hart, "Fast library for number theory: An introduction," in *Mathematical Software – ICMS 2010*, K. Fukuda, J. v. d. Hoeven, M. Joswig, and N. Takayama, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 88–91.
[19] S. Ahmad, C. Cachet, L. Demarest, B. Fuller, and A. Hamlin, "Proximity searchable encryption for the iris biometrics," Cryptology ePrint Archive, Report 2020/1174, 2020, https://ia.cr/2020/1174.

[20] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe, "FRVT 2006 and ICE 2006 large-scale experimental results," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 5, pp. 831–846, 2009.

[21] K. W. Bowyer and P. J. Flynn, "The ND-IRIS-0405 iris image dataset," *arXiv preprint arXiv:1606.04853*, 2016.

[22] P. J. Phillips, K. W. Bowyer, P. J. Flynn, X. Liu, and W. T. Scruggs, "The iris challenge evaluation 2005," in *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*. IEEE, 2008, pp. 1–8.

[23] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in *2012 IEEE 28th International Conference on Data Engineering*. IEEE, 2012, pp. 1156–1167.

[24] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604–613.

[25] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1. IEEE, 2005, pp. 172–183.

[26] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[27] E. M. Kornaropoulos, C. Papamanthou, and R. Tamassia, "Data recovery on encrypted databases with k-nearest neighbor query leakage," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1033–1050.

[28] E. M. Kornaropoulos and P. Efstathopoulos, "The case of adversarial inputs for secure similarity approximation protocols," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 247–262.

[29] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: ramification, attack and mitigation." in *NDSS*, vol. 20. Citeseer, 2012, p. 12.

[30] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 668–679.

[31] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill, "Generic attacks on secure outsourced databases," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1329–1340.

[32] G. Wang, C. Liu, Y. Dong, H. Pan, P. Han, and B. Fang, "Query recovery attacks on searchable encryption based on partial knowledge," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2017, pp. 530–549.

[33] P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart, "Leakage-abuse attacks against order-revealing encryption," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 655–672.

[34] P. Grubbs, M.-S. Lacharité, B. Minaud, and K. 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*, 2018, pp. 315–331.

[35] E. A. Markatou and R. Tamassia, "Full database reconstruction with access and search pattern leakage," in *International Conference on Information Security*. Springer, 2019, pp. 25–43.

[36] E. M. Kornaropoulos, C. Papamanthou, and R. Tamassia, "The state of the uniform: attacks on encrypted databases beyond the uniform query distribution," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1223–1240.

[37] F. Falzon, E. A. Markatou, D. Cash, A. Rivkin, J. Stern, and R. Tamassia, "Full database reconstruction in two dimensions," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 443–460.

[38] M. Barbosa, D. Catalano, A. Soleimanian, and B. Warinschi, "Efficient function-hiding functional encryption: From inner-products to orthogonality," in *Topics in Cryptology – CT-RSA 2019*, M. Matsui, Ed., 2019, pp. 127–148.

[39] P. Grubbs, A. Khandelwal, M.-S. Lacharité, L. Brown, L. Li, R. Agarwal, and T. Ristenpart, "Pancake: Frequency smoothing for encrypted data stores," in *29th USENIX Security Symposium*, 2020, pp. 2451–2468.

[40] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in *Theory of Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 457–473.

[41] K. Hollingsworth, K. W. Bowyer, and P. J. Flynn, "Similarity of iris texture between identical twins," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE, 2010, pp. 22–29.

[42] S. Ahmad and B. Fuller, "Unconstrained iris segmentation using convolutional neural networks," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 450–466.

[43] S. Ahmad. (2020) Sohaib ahmad github. Accessed: 2020-07-23. [Online]. Available: https://github.com/sohaib50k

[44] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern recognition*, vol. 43, no. 3, pp. 1016–1026, 2010.

[45] S. Simhadri, J. Steel, and B. Fuller, "Cryptographic authentication from the iris," in *International Conference on Information Security*. Springer, 2019, pp. 465–485.

[46] K. P. Hollingsworth, K. W. Bowyer, and P. J. Flynn, "The best bits in an iris code," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 964–973, 2008.

[47] R. M. Bolle, S. Pankanti, J. H. Connell, and N. K. Ratha, "Iris individuality: A partial iris model," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2. IEEE, 2004, pp. 927–930.

[48] P. Drozdowski, C. Rathgeb, A. Dantcheva, N. Damer, and C. Busch, "Demographic bias in biometrics: A survey on an emerging challenge," *IEEE Transactions on Technology and Society*, vol. 1, no. 2, pp. 89–103, 2020.

[49] S. Ziauddin and M. N. Dailey, "Iris recognition performance enhancement using weighted majority voting," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 277–280.

[50] Y. Kawai and K. Takashima, "Predicate- and attribute-hiding inner product encryption in a public key setting," in *Pairing-Based Cryptography – Pairing 2013*, Z. Cao and F. Zhang, Eds. Cham: Springer International Publishing, 2014, pp. 113–130.

[51] S. Ahmad, C. Cachet, L. Demarest, B. Fuller, and A. Hamlin. (2021) An implementation of proximity searchable encryption (PSE). https://github.com/chloecachet/pse.

[52] J. Akinyele, C. Garman, I. Miers, M. Pagano, M. Rushanan, M. Green, and A. Rubin, "Charm: A framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, 06 2013.

[53] K. Lewi. (2016) FHIPE github. https://github.com/kevinlewi/fhipe.

[54] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–5.

[55] J. Wang, H. Ma, Q. Tang, J. Li, H. Zhu, S. Ma, and X. Chen, "Efficient verifiable fuzzy keyword search over encrypted data in cloud computing." *Comput. Sci. Inf. Syst.*, vol. 10, no. 2, pp. 667–684, 2013.

[56] A. Boldyreva and N. Chenette, "Efficient fuzzy search on encrypted data," in *International Workshop on Fast Software Encryption*. Springer, 2014, pp. 613–633.

[57] K. Zhou and J. Ren, "Passbio: Privacy-preserving user-centric biometric authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 3050–3063, 2018.

[58] A. Boldyreva and T. Tang, "Privacy-preserving approximate k-nearest-neighbors search that hides access, query and volume patterns," *PoPETS Proceedings on Privacy Enhancing Technologies*, 2021.

[59] S. Kamara, T. Moataz, and O. Ohrimenko, "Structured encryption and leakage suppression," in *CRYPTO*. Springer, 2018, pp. 339–370.

[60] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: The power of file-injection attacks on searchable encryption," in *25th USENIX Security Symposium*, 2016, pp. 707–720.

[61] M. Lacharité, 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)*, 2018, pp. 297–314.

[62] C. Evrendilek and H. Akcan, "On the complexity of trilateration with noisy range measurements," *IEEE Communications Letters*, vol. 15, no. 10, pp. 1097–1099, 2011.

[63] R. C. Tillquist, R. M. Frongillo, and M. E. Lladser, "Metric dimension," *arXiv preprint arXiv:1910.04103*, 2019.

[64] L. Laird, R. C. Tillquist, S. Becker, and M. E. Lladser, "Resolvability of Hamming graphs," *SIAM Journal on Discrete Mathematics*, vol. 34, no. 4, pp. 2063–2081, 2020.

[65] L. Laird, "Metric dimension of hamming graphs and applications to computational biology," *arXiv preprint arXiv:2007.01337*, 2020.

A LEAKAGE 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 [59]. 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 run attacks to recover either the queries being made or the data stored in the database. We consider attacks that rely on injecting files or queries [60] to be out of scope. Common, attackable, relevant leakage profiles are:

(1) *Response length leakage* [31, 34] 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* [32] the attacker is able to glean which queries are querying the same value, but not necessarily the value itself. Attacks on this profile rely on having information about the query distribution, and much like the response length leakage attacks, match with that auxiliary information based on frequency.

(3) *Access pattern leakage* [29, 30] here the attacker is given knowledge if the same dataset element is returned for different 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 [33, 34, 36, 37, 61]. Building on the co-occurrence matrix (available with access pattern leakage) consider the case when records a, b, c are returned by a first query and c, d 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 collect an ordering of all records (up to reflection).

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 $\mathcal{D}(x_i, y)$ and is trying to find the location of y. Determining the location of y requires k to be one larger than the dimension. The problem is more difficult but well studied for approximate distances [62]. Similar ideas can be applied in discrete metrics with each learned distance reducing the set of possible y. In the Hamming metric of dimension n, $k = \Theta(n)$ suffices [63–65].

B ADDITIONAL STATISTICAL ANALYSIS

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 6 shows the FAR/FRR tradeoff 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.

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 6: TAR for different 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.

C PROOF OF THEOREM 1

This scheme has the security as the original $\mathsf{IPE}_{\mathsf{fh,sk,pred}}$ scheme from [38] for the simulation based security definition. We note that that scheme of Barbosa et al. [38] builds on the work Kim et al.[12] and our proof uses similar definitions of formal variables. The scheme works by having a challenger interact with a simulator S and two oracles, O'_{TokGen} and O'_{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 $\mathcal{S}$ 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'_{TokGen} and O'_{Encrypt} to match the functionality of the encryption scheme.

Inner-product collection: Let i, j be shared counters between the token generation and encryption oracles. Let $x^{(i)} \in \mathbb{Z}_q^n$ and $y^{(j)} \in \mathbb{Z}_q^n$ 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

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

Formal variables: The simulator constructs formal variables for the unknowns of the system in order to respond as correctly as possible. Let Q be the maximum number of queries made by an adversary. Let σ and N be as in the construction in Figure 4. For all $i \in [Q]$, $\ell \in [\sigma]$ and $k \in [N]$, let $\hat{\alpha}^{(i)}, \hat{\beta}^{(i)}, \hat{x}^{(i)}_{\ell,k}, \hat{y}^{(i)}_{\ell,k}$ represent the hidden variables $\alpha^{(i)}, \beta^{(i)}, x^{(i)}_{\ell,k}, y^{(i)}_{\ell,k}$, let $\hat{b}_{\ell,k,m}$ and $\hat{b}^*_{\ell,k,m}$ represent the entry in position (k, m) of the $\mathbb{B}_\ell$ and $\mathbb{B}^*_\ell$ matrices respectively, let $\hat{\zeta}^{(i)}_\ell$ be the formal variables for $\zeta^{(i)}_\ell$ where the simulator tracks the constraints that for each $i \in [Q]$, $\sum_{\ell=1}^{\sigma} \hat{\zeta}^{(i)}_\ell = 0$ and let $\hat{s}^{(i)}_{\ell,m}$ and $\hat{t}^{(i)}_{\ell,m}$ represent formal polynomials as constructed below,

$$\hat{s}^{(i)}_{\ell,m} = \sum_{k=1}^{N} \hat{y}'^{(i)}_{\ell,k} \cdot \hat{b}_{\ell,k,m} = \hat{b}_{\ell,1,m} + \sum_{k=2}^{N} \hat{y}^{(i)}_{\ell,k-1} \cdot \hat{b}_{\ell,k,m} \tag{1}$$

$$\hat{t}^{(i)}_{\ell,m} = \sum_{k=1}^{N} \hat{x}'^{(i)}_{\ell,k} \cdot \hat{b}^*_{\ell,k,m} = \hat{\zeta}^{(i)}_\ell \cdot \hat{b}^*_{\ell,1,m} + \sum_{k=2}^{N} \hat{x}^{(i)}_{\ell,k-1} \cdot \hat{b}^*_{\ell,k,m} \tag{2}$$

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

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

and

$$\mathcal{T} = \left\{\hat{\alpha}^{(i)}, \hat{\beta}^{(i)}\right\}_{i\in[Q]} \cup \left\{\hat{x}'^{(i)}_{\ell,k},\ \hat{y}'^{(i)}_{\ell,k},\ \hat{\zeta}^{(i)}_\ell\right\}_{i\in[Q],\ell\in[\sigma],k\in[N]} \cup \left\{\hat{b}_{\ell,k,m},\ \hat{b}^*_{\ell,k,m}\right\}_{\ell\in[\sigma],\ m,k\in[N]}$$

Specification of the simulator Let $\mathcal{A}$ be a PPT adversary that makes at most $Q = \mathsf{poly}(\lambda)$ queries to the oracles. The simulator $\mathcal{S}$ starts by initializing an empty set of inner products C_{ip} and three empty tables T_1, T_2, T_T which map handles to the polynomials over the variables of $\mathcal{R}$. The state of the simulator consists of

these four objects, $(C_{\mathsf{ip}}, T_1, T_2, T_T)$, which are updated after each query received. The simulator $\mathcal{S}$ answers the adversary's queries as follows.

Token generation queries: On input $x^{(i)} \in Z_q^n$, O'_{TokGen} sends the collection C'_{ip} to the simulator. $\mathcal{S}$ updates $C_{\mathsf{ip}} \leftarrow C'_{\mathsf{ip}}$. For $1 \le \ell \le \sigma$, $1 \le m \le N$, $\mathcal{S}$ generates a new handle $h_{\ell,m} \xleftarrow{\$} \{0,1\}^\lambda$ and adds the mapping $h_{\ell,m} \to \hat{\alpha}^{(i)} \cdot \hat{s}^{(i)}_{\ell,m}$ to T_1. $\mathcal{S}$ then sets $\mathsf{tk}_\ell = h_{\ell,1}, \cdots, h_{\ell,N}$. Finally, $\mathcal{S}$ returns the token $\mathsf{tk} = (\mathsf{tk}_1, \cdots, \mathsf{tk}_\sigma)$.

Encryption queries: On input $y^{(i)} \in Z_q^n$, O'_{Encrypt} sends the collection C'_{ip} to the simulator. $\mathcal{S}$ updates $C_{\mathsf{ip}} \leftarrow C'_{\mathsf{ip}}$. For $1 \le \ell \le \sigma$, $1 \le m \le N$, $\mathcal{S}$ generates a new handle $h_{\ell,m} \xleftarrow{\$} \{0,1\}^\lambda$ and adds the mapping $h_{\ell,m} \to \hat{\beta}^{(i)} \cdot \hat{t}^{(i)}_{\ell,m}$ to T_2. $\mathcal{S}$ sets $\mathsf{ct}_\ell = h_{\ell,1}, \cdots, h_{\ell,N}$. Finally, $\mathcal{S}$ returns the ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \cdots, \mathsf{ct}_\sigma)$.

Addition oracle queries: Given $h_1, h_2 \in \{0,1\}^\lambda$, $\mathcal{S}$ verifies that formal polynomials p_1, p_2 exist in table T_τ, $\tau \in \{1, 2, T\}$ such that $h_1 \to p_1$ and $h_2 \to p_2$. If it is not the case $\mathcal{S}$ returns $\perp$. If a handle for (p_1+p_2) already exists in T_τ $\mathcal{S}$ returns it. Otherwise, $\mathcal{S}$ generates a new handle $h \xleftarrow{\$} \{0,1\}^\lambda$, adds the mapping $h \to (p_1 + p_2)$ to T_τ and returns h.

Pairing oracle queries: Given $h_1, h_2 \in \{0,1\}^\lambda$, $\mathcal{S}$ verifies that formal polynomials p_1, p_2 exist in tables T_1 and T_2 respectively, such that $h_1 \to p_1$ in T_1 and $h_2 \to p_2$ in T_2. If it is not the case $\mathcal{S}$ returns $\perp$. If a handle for $(p_1 \cdot p_2)$ already exists in T_T $\mathcal{S}$ returns it. Otherwise, $\mathcal{S}$ generates a new handle $h \xleftarrow{\$} \{0,1\}^\lambda$, adds the mapping $h \to (p_1 \cdot p_2)$ to T_T and returns h.

Zero-testing oracle queries: Given $h \in \{0,1\}^\lambda$, $\mathcal{S}$ verifies that formal polynomials p exists in T_τ, $\tau \in \{1, 2, T\}$, such that $h \to p$. If it is not the case $\mathcal{S}$ returns $\perp$. $\mathcal{S}$ then works as follows.

(1) It "canonicalizes" the polynomial p by expressing it as a sum of products of formal variables in $\mathcal{T}$ with $\mathtt{poly}(\lambda)$ terms.
(2) If $\tau \in \{1, 2\}$ and p is the zero polynomial, $\mathcal{S}$ outputs "zero". Otherwise if outputs "non-zero".
(3) If $\tau = T$ the simulator decomposes p into the form

$$p = \sum_{i,j=1}^{Q} \hat{\alpha}^{(i)} \hat{\beta}^{(j)} \cdot \left(p_{i,j}\left(\left\{ \hat{s}^{(i)}_{\ell,m},\ \hat{t}^{(j)}_{\ell,m} \right\}_{\ell \in [\sigma], m \in [N]} \right) + f_{i,j}\left(\left\{ \hat{s}^{(i)}_{\ell,m},\ \hat{t}^{(j)}_{\ell,m} \right\}_{\ell \in [\sigma], m \in [N]} \right) \right) \tag{3}$$

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

$$p_{i,j} = c_{i,j} \cdot \left(\sum_{\ell,m=1}^{\sigma,N} \hat{s}^{(i)}_{\ell,m}\ \hat{t}^{(j)}_{\ell,m} \right)$$

where $c_{i,j} \in \mathbb{Z}_q$ is the coefficient of the term $\hat{s}^{(i)}_{1,1} \hat{t}^{(j)}_{1,1}$, and $f_{i,j}$ consists of the remaining terms.
(4) If for all $1 \le i, j \le Q$, $(i, j) = 0$ in C_{ip} (corresponding to a zero inner product) and $f_{i,j}$ does not contain any non-zero term, $\mathcal{S}$ outputs "zero". Otherwise it outputs "non-zero".

Correctness of the simulator As in the original proof, the simulator's responses to token generation, encryption and group oracle queries are distributed identically as in the real experiment. We now have to show correctness of the simulator's answers to zero-testing oracle queries.

(1) We first need to show that the canonicalization process in step 1 is efficient. Since the adversary can only obtain handles to new monomials using token generation and encryption queries, the monomials are all over formal variables in $\mathcal{R}$. 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 $\mathtt{poly}(Q)$ terms and degree 2.
Then using Equations 1 and 2, the formal polynomial p can be expressed as a polynomial over formal variables in $\mathcal{T}$. Since p has degree at most 2 over variables in $\mathcal{R}$, it can be expressed as a sum of at most $\mathtt{poly}(Q, n)$ monomials over variables in $\mathcal{T}$ and has degree at most $\mathtt{poly}(n)$. Since both the polynomial over $\mathcal{R}$ and the canonical polynomial over $\mathcal{T}$ are polynomially-sized, this is efficient.
(2) For $\tau = 1$, the only monomials the adversary can obtain are responses to token generation queries. Then the canonical polynomial is of the form

$$\begin{aligned} p &= \sum_{i=1}^{Q} \hat{\alpha}^{(i)} \left(\sum_{\ell,m=1}^{\sigma,N} c^{(i)}_{\ell,m} \cdot \hat{s}^{(i)}_{\ell,m} \right) \\ &= \sum_{i=1}^{Q} \hat{\alpha}^{(i)} \left(\sum_{\ell,m=1}^{\sigma,N} c^{(i)}_{\ell,m} \sum_{k=1}^{N} \hat{y}'^{(i)}_{\ell,k} \cdot \hat{b}_{\ell,k,m} \right) \\ &= \sum_{i=1}^{Q} \hat{\alpha}^{(i)} \left(\sum_{\ell,m=1}^{\sigma,N} c^{(i)}_{\ell,m} \left(\hat{b}_{\ell,1,m} + \sum_{k=2}^{N} \hat{y}^{(i)}_{\ell,k} \cdot \hat{b}_{\ell,k,m} \right) \right) \end{aligned}$$

where $c^{(i)}_{1,1}, \cdots, c^{(i)}_{\sigma,N} \in \mathbb{Z}_q$.
Notice that the sum $\hat{b}_{\ell,1,m} + \sum_{k=2}^{N} \hat{y}^{(i)}_{\ell,k} \cdot \hat{b}_{\ell,k,m}$ can never be the identically zero polynomial over the formal variables $\{\hat{b}_{\ell,k,m}\}_{\ell \in [\sigma],\ k,m \in [N]}$. This holds irrespective of the actual values of the adversary's query $x^{(i)}$. Since all $\{\hat{\alpha}^{(i)}\}_{i \in [Q]}$ and $\{\hat{b}_{\ell,k,m}\}_{\ell \in [\sigma],\ k,m \in [N]}$ are sampled uniformly and independently in the real game and the polynomial p has degree $\mathtt{poly}(n) = \mathtt{poly}(\lambda)$, then by the Schwartz-Zippel lemma [12, Lemma 2.9], p evaluates to non-zero with overwhelming probability. This implies that the simulator is correct with overwhelming probability.
(3) For $\tau = 2$, the only monomials the adversary can obtain are responses to ciphertexts queries. Then the canonical polynomial is of the form

$$\begin{aligned} p &= \sum_{i=1}^{Q} \hat{\beta}^{(i)} \left(\sum_{\ell,m=1}^{\sigma,N} c^{(i)}_{\ell,m} \cdot \hat{t}^{(i)}_{\ell,m} \right) \\ &= \sum_{i=1}^{Q} \hat{\beta}^{(i)} \left(\sum_{\ell,m=1}^{\sigma,N} c^{(i)}_{\ell,m} \sum_{k=1}^{N} \hat{x}'^{(i)}_{\ell,k} \cdot \hat{b}^{*}_{\ell,k,m} \right) \\ &= \sum_{i=1}^{Q} \hat{\beta}^{(i)} \left(\sum_{\ell,m=1}^{\sigma,N} c^{(i)}_{\ell,m} \left(\zeta^{(i)}_{\ell} \cdot \hat{b}^{*}_{\ell,1,m} + \sum_{k=2}^{N} \hat{x}^{(i)}_{\ell,k} \cdot \hat{b}^{*}_{\ell,k,m} \right) \right) \end{aligned}$$

where $c^{(i)}_{1,1}, \cdots, c^{(i)}_{\sigma,N} \in \mathbb{Z}_q$. Notice that the sum $\zeta^{(i)}_{\ell} \cdot \hat{b}^{*}_{\ell,1,m} + \sum_{k=2}^{N} \hat{x}^{(i)}_{\ell,k} \cdot \hat{b}^{*}_{\ell,k,j}$ can only be the identically zero polynomial

over the formal variables $\{\hat{b}^*_{\ell,k,m}\}_{\ell\in[\sigma],\, k,m\in[N]}$ if $\zeta_\ell^{(i)} = 0$ which happens with negligible probability. Again, this holds irrespective of the adversary's queries $y^{(1)}, \cdots, y^{(Q)}$ and p is not the identically zero polynomial over the formal variables $\{\hat{\beta}^{(i)}\}_{i\in[Q]}$ and $\{\hat{b}^*_{\ell,k,m}\}_{\ell\in[\sigma],\, k,m\in[N]}$. Since all $\hat{b}^*_{\ell,k,m}$ are independent from one another (since $\hat{b}_{\ell,k,m}$ was sampled uniformly and independently), then again by Schwartz-Zippel lemma p evaluates to non-zero with overwhelming probability, the simulator is correct with overwhelming probability.

(4) For $\tau = T$, the only polynomials the adversary can obtain are products of two polynomials, one from each base group. Then the polynomial p can be decomposed into a sum of monomials that each contain $\alpha^{(i)}$ and $\beta^{(j)}$ for some $i, j \in [Q]$. Then $\mathcal{S}$ can regroup terms for each $i, j \in [Q]$ and obtain Equation 3. If $f_{i,j}$ does not contain any term, then p is of the form

$$
\begin{aligned}
p &= \sum_{i,j=1}^{Q} \hat{\alpha}^{(i)}\hat{\beta}^{(j)} \cdot c_{i,j} \cdot \left(\sum_{\ell,m=1}^{\sigma,N} \hat{s}_{\ell,m}^{(i)}\, \hat{t}_{\ell,m}^{(j)} \right) \\
&= \sum_{i,j=1}^{Q} \hat{\alpha}^{(i)}\hat{\beta}^{(j)} \cdot c_{i,j} \cdot \left(\sum_{\ell,m=1}^{\sigma,N} \left(\sum_{k=1}^{N} \hat{y}'^{(i)}_{\ell,k} \cdot \hat{b}_{\ell,k,m} \right) \right. \\
&\quad \left. \cdot \left(\sum_{k=1}^{N} \hat{x}'^{(j)}_{\ell,k} \cdot \hat{b}^*_{\ell,k,m} \right) \right) \\
&= \sum_{i,j=1}^{Q} \hat{\alpha}^{(i)}\hat{\beta}^{(j)} \cdot c_{i,j} \cdot \left(\sum_{\ell=1}^{\sigma} (\hat{x}'^{(j)}_{\ell})^T \cdot \mathbb{B}^*_\ell \cdot \mathbb{B}^T_\ell \cdot \hat{y}'^{(i)}_{\ell} \right) \\
&= \sum_{i,j=1}^{Q} \hat{\alpha}^{(i)}\hat{\beta}^{(j)} \cdot c_{i,j} \cdot \left(\sum_{\ell=1}^{\sigma} \zeta_\ell^{(i)} + \langle \hat{x}_\ell^{(j)}, \hat{y}_\ell^{(i)} \rangle \right) \\
&= \sum_{i,j=1}^{Q} \hat{\alpha}^{(i)}\hat{\beta}^{(j)} \cdot c_{i,j} \cdot \langle \hat{x}^{(j)}, \hat{y}^{(i)} \rangle
\end{aligned}
$$

p is the zero polynomial when all (i, j) inner products are zero, which can be known by checking if $(i, j) \to 0$ in C_{ip}. 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 $\{\, \hat{b}_{\ell,k,m} \,\}_{\ell\in[\sigma],k,m\in[N]}$, irrespective of the adversary's choice of admissible queries. We refer the reader to the original work [12, Section 3] for a detailed proof of this claim. Then by the Schwartz-Zippel lemma, p evaluates to non-zero with overwhelming probability when $f_{i,j}$ contains at least one term.

D 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 $\mathcal{D}(y, x_i) \le t$ with $\mathcal{D}(y, x_i)$ the Hamming distance between vectors y and x_i. Then there exists a unique $\mathsf{tk}_j \in Q_{y,t}$ such that $b_j \leftarrow \mathsf{IPE.Decrypt}(\mathsf{pp}, \mathsf{tk}_j, \mathsf{ct}_i)$ 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 $\mathsf{tk}_j \in Q_{y,t}$, $b_j \leftarrow \mathsf{IPE.Decrypt}(\mathsf{pp}, \mathsf{tk}_j, \mathsf{ct}_i)$ and $b_j = 1$ with negligible probability. Then considering the worst case where either $\mathcal{D}(y, x_\ell) = t$ or for all $x_i \in X$, $f_{y,t}(x_i) = 0$, we have:

$$
\begin{aligned}
&\Pr\left[\mathsf{PSE.Search}(\mathsf{pp}, Q_{y,t}, I_X) = J_{X,y,t}\right] \\
&\quad \ge 1 - \ell(t+1) \times \Pr\left[\begin{matrix} \mathsf{IPE.Decrypt}(\mathsf{pp},\mathsf{tk}_j,\mathsf{ct}_i) \\ \neq (\mathcal{D}(x_i,y) \stackrel{?}{=} d_j) \end{matrix} \right] \\
&\quad \ge 1 - \ell(t+1) \times \mathsf{negl}(\lambda).
\end{aligned}
$$

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

(1) $\mathcal{A}_{\mathsf{IPE}}$ receives pp from C_{IPE} and forwards it to $\mathcal{A}_{\mathsf{PSE}}$.
(2) $\mathcal{A}_{\mathsf{IPE}}$ receives two m-query histories $\mathsf{History}^{(0)}, \mathsf{History}^{(1)}$ from $\mathcal{A}_{\mathsf{PSE}}$ where $\mathsf{History}^{(\beta)} = (X^{(\beta)}, F^{(\beta)})$ for $\beta \in \{0, 1\}$.
(3) For each $x_i^{(\beta)} \in X^{(\beta)}$, $i \in [1, \ell]$, $\mathcal{A}_{\mathsf{IPE}}$ encodes it as $x_i^{(\beta)*} \in \{\text{-}1, 1\}^n$ and creates the query $S_i = (x_i^{(0)*} \| \text{ -}1,\ x_i^{(1)*} \| \text{ -}1)$.
(4) $\mathcal{A}_{\mathsf{IPE}}$ sets $S = S_1, \cdots, S_\ell$.
(5) For each $f_j^{(\beta)} \in F^{(\beta)}$, $j \in [1, m]$:
  (a) $\mathcal{A}_{\mathsf{IPE}}$ extracts a vector $y_j^{(\beta)} \in \mathbb{Z}_q^n$ and $t \in \mathbb{N}$.
  (b) $\mathcal{A}_{\mathsf{IPE}}$ encodes $y_j^{(\beta)}$ as $y_j^{(\beta)*} \in \{\text{-}1, 1\}^n$ and creates $D_j^{(0)} = (d_0, \cdots, d_t)$ such that $d_k = n - 2k$ with $0 \le k \le t$.
  (c) $\mathcal{A}_{\mathsf{IPE}}$ 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 $\left(\langle x_i^{(0)}, y_j^{(0)} \rangle \stackrel{?}{=} d_k^{(0)}\right) = \left(\langle x_i^{(1)}, y_j^{(1)} \rangle \stackrel{?}{=} d_k^{(1)}\right)$. ($\mathcal{A}_{\mathsf{IPE}}$ can always find a permutation to make this last condition by the admissibility requirement.)
  (d) $\mathcal{A}_{\mathsf{IPE}}$ samples a random permutation $\psi_j : [0, t] \to [0, t]$.
  (e) For $0 \le k \le t$, $\mathcal{A}_{\mathsf{IPE}}$ creates $y_j^{(\beta)*} \| d_k^{(\beta)}$ with $\beta \in \{0, 1\}$, $d_k^{(0)} \in D_j^{(0)*}$ and $d_k^{(1)} \in D_j^{(1)}$. Then $\mathcal{A}_{\mathsf{IPE}}$ computes
  $$R_j^{(\beta)} = \psi_j\left(y_j^{(\beta)*} \| d_0^{(\beta)}, \cdots, y_j^{(\beta)*} \| d_t^{(\beta)} \right)$$
  and sets $R_j = (R_j^{(0)}, R_j^{(1)})$.
  (f) $\mathcal{A}_{\mathsf{IPE}}$ sets $R = R_1, \cdots, R_m$.
(6) $\mathcal{A}_{\mathsf{IPE}}$ sends the token generation queries R and encryption queries S to C_{IPE} and receives back a set of tokens $T^{(\beta)} = \mathsf{tk}_{1,0}^{(\beta)}, \cdots, \mathsf{tk}_{m,t}^{(\beta)}$ and a set of encrypted keywords $C^{(\beta)} = \mathsf{ct}_1^{(\beta)}, \cdots, \mathsf{ct}_\ell^{(\beta)}$ such that
$$\mathsf{tk}_{j,k}^{(\beta)} \leftarrow \mathsf{IPE.TokGen}(\mathsf{sk},\ y_j^{(\beta)*} \| d_k^{(\beta)})$$
$$\mathsf{ct}_i^{(\beta)} \leftarrow \mathsf{IPE.Encrypt}(\mathsf{sk},\ x_i^{(\beta)*} \| \text{ -}1)$$
for $i \in [1, \ell]$, $j \in [1, m]$, $k \in [0, t]$ and $\beta \in \{0, 1\}$. $\mathcal{A}_{\mathsf{IPE}}$ forwards $T^{(\beta)}$ and $C^{(\beta)}$ to $\mathcal{A}_{\mathsf{PSE}}$, respectively as the encrypted index $I^{(\beta)}$ and the list of queries $Q^{(\beta)}$.
(7) $\mathcal{A}_{\mathsf{IPE}}$ receives $\beta' \in \{0, 1\}$ from $\mathcal{A}_{\mathsf{PSE}}$ and returns it.

Since the number of token generation queries, $m \times t$, sent by $\mathcal{A}_{\mathsf{IPE}}$ remains polynomial in the security parameter, the advantage of $\mathcal{A}_{\mathsf{PSE}}$ is

$$\mathsf{Adv}^{\mathsf{Exp}^{\mathsf{PSE}}_{IND}}_{\mathcal{A}_{\mathsf{PSE}}} = \mathsf{Adv}^{\mathsf{Exp}^{\mathsf{IPE}}_{IND}}_{\mathcal{A}_{\mathsf{IPE}}}$$

This completes the proof of Theorem 2.