SaTC: CORE: Small: Collaborative:
Next-Generation Secure Outsourced Databases

1 Introduction

The importance of collecting and storing data is universal, with use cases in governmental [PB14], commercial [LB02, ins15], and personal sectors [MvHC+11]. Storing and querying large datasets has tremendous value in improving decision making, but this growth in size and complexity is increasingly resulting in organizations relying on external cloud providers for their data needs.

This outsourced storage represents a natural attack target. Attacks occur against both government [Tim15b] and commercial [Tim15a, Gre17] datasets. One natural response to this risk is to encrypt data before outsourcing. However, employing encryption comes at the cost of disabling the cloud server from *quickly processing the data and answering complex queries from the client*. Ideally, we could use more sophisticated cryptographic techniques to *create databases capable of efficiently answering a client's queries without revealing information to the cloud server*.

Numerous advanced cryptographic techniques have been proposed to achieve this goal [PR12, CGKO06, CGN98, BDOP04]. This research has largely split into two research threads: property-preserving encryption (PPE) which emphasizes background compatibility and use of legacy database management systems, and searchable secure encryption (SSE) which emphases security.

PPE creates symmetric encryption techniques compatible with an unprotected database. Examples include deterministic encryption [BBO07], which can answer equality queries, and order-preserving encryption [BCO11, BCLO09], which can answer range queries. Academic teams, start-up companies (including Bitglass, Ciphercloud, Crypteron, PreVeil, Skyhigh, ZeroDB) and Fortune 500 companies (including Microsoft's SQL Server 2016 and Azure and Google's Encrypted BigQuery) offer variants of property-preserving encryption.

By contrast, SSE schemes can be viewed as starting from secure multi-party computation and optimizing solutions for common database tasks (both [FVK+15] and [IKLO16] explicitly use multi-party computation for sensitive subcomputations). This approach requires redesign of database indexing mechanisms and achieves better security at the cost of decreased efficiency and backward compatibility.

These systems have been implemented at moderate scale. Both property-preserving solutions [PRZB12, PBP16] and searchable encryption [PKV+14, FVK+15, CJJ+13, JJK+13, CJJ+14, FJK+15, IKLO16] solutions have been tested on datasets with billions of records.

1.1 Use Cases

Emerging databases include large scale graph databases, analytic databases, and biometric databases.

1. Many data sets are naturally interpreted as large sparse graphs. Examples include social networks such as Facebook and Twitter and communities such as organizational communication, academic co-authorship, and the co-stardom network. This type of network is also used to perform Internet scale analysis. Common graph algorithms include computing triangles (sets of nodes $\{a, b, c\}$ where all pairs are close according to a metric), shortest path algorithms, network diameter, and degree distribution.

2. Increasingly, databases are not asked to return subsets of data but rather derive statistics and analytics about the stored data. Machine-learning-as-a-service has emerged as a business model for large and small companies [Azu17]. Many machine learning algorithms depend on computing distance between points. For example, linear regression finds the line that minimizes the sum of distances between the line and data points. Similarly, the first principal component minimizes the sum of distances between the selected line and the dataset [WEG87].

3. The FBI has long held a fingerprint database with hundreds of millions of records [BBOH96] for identifying criminals. Increasingly, countries are using biometrics as an identifier for citizens, linking biometrics with unique identifiers. The Aadhaar system in India links biometrics with a unique 12 digit number with over 1 billion numbers issued [Dau14]. Increasingly, passports are biometric enabled [Sta08]. As these databases move to indexing all citizens, privacy concerns abound. These databases compare the distance between a target point a and the set of stored points b_i, returning all points within some defined threshold distance from the target.

These databases share a fundamental operation: *computing distance/proximity of tuples of points*.

1.2 Inadequacy of Prior Work

In 2000, Song, Wagner, and Perrig provided the first scheme with communication proportional to the description of the query and the server performing (roughly) a linear scan of the encrypted database [SWP00]. There has been tremendous work since 2000: both PPE and SSE approaches handle much of SQL and NoSQL queries and scale to datasets of billions of records. Though there has been some work on computing shortest paths [MKNK15], in general neither PPE nor SSE is capable of handling geometric data that is prominent in the applications discussed above. Furthermore, each approach also has a second weakness:

1. PPE has been subject to a number of leakage-abuse attacks that show that order-preserving encryption (and sometimes deterministic encryption) is not safe in most cases [NKW15, CGPR15, KKNO16, PW16, GMN$^+$16, GSB$^+$16, ZKP16]. Nonetheless, industry has primarily adopted this approach.

2. SSE solutions are more limited in functionality and efficiency than PPE based solutions. The fastest SSE based solutions report overhead of 300% [CJJ$^+$13, JJK$^+$13, CJJ$^+$14, FJK$^+$15] while PPE based solutions report overhead of 30% [PRZB12]. We are only aware of a single SSE solution that can handle JOIN statements [KM16]. As of this writing, the information leaked by this scheme is not clear. These solutions replace the entire database software stack with a custom "cryptographic" database. We posit that the lack of backward compatibility and administrative drawbacks hurt industry adoption.

The emergence of PPE systems indicates the approach has benefits beyond efficiency. These benefits include backward compatibility, use of legacy software, and improved transparency. The proposed research will consider proximity queries as a case study to understand the following question.

> *Is it possible to securely outsource modern databases while using a traditional database software stack? What is the minimum amount that an unprotected database must be modified to achieve security?*

Out of scope. We do not address approaches based on fully-homomorphic encryption [Gen09] or functional encryption [GGH$^+$13], which are still too slow to be practical for the data sets we are considering. In addition, we do not consider improvements to private information retrieval [CGKS95] or oblivious

RAM [Gol87, GO96] to be in scope for this proposal, though we may make use of these constructions within our schemes. We will also not consider developing new leakage inference attacks. However, the PIs are well aware of (and have contributed to) to these attacks on secure databases, and they motivate the solutions we are proposing.

Lastly, secure enclaves such as Intel SGX offer a promising hardware approach that is being used to isolate programs and provide security [CD16]. SGX can be used to simulate cryptographic primitives [SGF17, FVBG16]. We believe that SGX can be used to upgrade the security of secure databases from honest-but-curious to malicious. However, we leave combining our techniques with secure hardware to future work.

1.3 Proposed Work

The goal of this proposal is to create secure outsourced databases that will be adopted and used by industry. Towards achieving this goal we recognize two lessons from the past: (1) PPE must be carefully analyzed to understand security and (2) there are tremendous benefits of maintaining backwards compatibility with existing database systems. Together, these lessons tell us that some modification of database systems may be necessary but these modifications should be judicious. This will be our guiding principle thorough this project. Our goal is to create a third approach to secure outsourced databases:

> *Cryptographic operations are restricted to only database operators (used to create index structures). These cryptographic operators can call interactive protocols but do not require replacing the unprotected database.*

The proposed research will consider proximity queries as a case study and give constructions supporting these queries. The research will be split into three components:

Section 2. To show the promise of this approach, we will first consider whether property-preserving encryption can be used to answer proximity queries. We expect that the strength and utility of *distance-preserving encryption* will depend on the definition of security, in particular questions such as: (1) should distance be precisely preserved? (2) is distance-revealing encryption sufficient? and (3) what is the distance metric? Building on the research of the PI on order-preserving encryption [BCLO09, BCO11], this component will consist of the following tasks:

1. Analysis of security provided by distance-preserving and distance-comparison preserving encryption for common metrics
2. Design of distance-revealing encryption
3. Design of approximate distance-revealing encryption

Section 3. When the security provided by distance-revealing encryption appears inadequate for an application, we will change the model to allow operators to call interactive protocols. We will show what functionality can be built using *interactive operators*, drawing on partial order-preserving encryption (POPE), recent work of the co-PIs [PKV$^+$14, RACY16]. This component will consist of the following tasks:

1. Developing *Partial Distance-Preserving Encryption* for Euclidean and distance proximity queries
2. Adding forward security to POPE via re-insertions and random shuffling
3. Enabling multidimensional queries in BlindSeer and removing the need for additive homomorphic encryption and secure two-party computation in the outsourced storage setting

4. Judiciously incorporating oblivious RAM (ORAM) to reduce leakage
5. Developing a new hybrid approach that uses approximate distance-revealing encryption and interactive protocols to obtain better performance and backwards compatibility

Section 4. Throughout the project, we will implement and analyze our most promising schemes and place them in a fair context with prior and ongoing work by others. The goal will be to give practitioners as well as researchers a clear and *consistent* picture of the tradeoffs between security, performance, and backwards compatibility. We will use the UConn HPC Cluster to evaluate our implementations on realistic-size databases (`https://hpc.uconn.edu`). We expect this evaluation to use and extend the database and query generator created as part of the IARPA SPAR project [VPH+15]. The co-PI oversaw development of the most recent version of these tools. Specifically, our tasks are:

1. Analyzing and developing metrics for the leakage of information or access patterns inherent in our new PPE and interactive schemes
2. Developing and providing open-source access to the new tools that we develop
3. Examining the performance of our approaches on realistic and repeatable experiments, using industrially-relevant datasets and query benchmarks

1.4 Intellectual Merits and Broader Impacts

Intellectual merits. Encrypted databases and searchable encryption have rich histories rooted in the design of oblivious random access machines. The field has been the focus of multiple large scale projects including IARPA's APP and SPAR [IAR11]. The field is quite diverse bringing together cryptographers, system researchers, and database experts. Furthermore, there is clear demand in industry for solutions. Data breaches are becoming nearly daily events. However, recent leakage inference attacks have taught us that just because something is called encrypted does not make it secure. This problem requires careful design that balances functionality, efficiency, and security needs. The rapid deployment of property-preserving techniques has reinforced the importance of simplicity, efficiency and backward compatibility. These two developments inform the core of our approach: using interactivity to strengthen property-preserving encryption.

Broader impacts. The confidentiality of data is a core societal tenant. Deployed encrypted databases provide little security and may even hurt by providing a false sense of security. There is a tremendous need for research in this area to understand the tradeoffs between security, functionality, and efficiency.

The PIs are committed to broad dissemination of research material. The PIs have participated in large scale evaluation of searchable encryption, working on both constructions and attacks. The PIs have a track record of releasing the source code implementations of their work (including previous work on searchable encryption) and contributing to open-source projects. The protocols and other products of this proposal will be released publicly as open-source implementations.

Lastly, all PIs are dedicated to engaging with undergraduates. Two co-PIs being at an undergraduate only institution, the US Naval Academy, where co-PI Roche was recently recognized with the institution-wide Apgar Award for Teaching. PI O'Neill has previously been awarded an REU and used it to work with a female undergraduate student. Co-PI Fuller supervises the cyber-security club at the University of Connecticut and supports their efforts to understand and research computer security. This club allows students with varying educational preparation to engage outside of the classroom and learn the impact of computer science and security.

2 Property-Preserving Encryption for Proximity

In spatial databases, nearest neighbor queries (*e.g.*, finding the closest soldier in the field) and clustering queries are pervasive. In large scale biometric databases, the fundamental operation is comparison of the distance between a target point and all stored points. Often, biometric databases return the nearest match if the distance is less than some threshold. In this component of the proposal, we will consider encryption algorithms which support various types of proximity queries.

Core operations. Despite the varied use of proximity functionality in databases, we find two core operations. The first core operation is computing the distance between two points a and b or $d(a, b)$ according to some metric d. This distance can be granular, for example $d(a, b) = 15$ or coarse, $d(a, b) \in \{0, 1, 2\}$ corresponding to $\{equal, close, far\}$. Even with this small set of outputs, users expect the function d to behave as a metric so we restrict our attention to metric functions. We call property-preserving encryption where $d(\mathsf{Enc}(a), \mathsf{Enc}(b)) = d(a, b)$ *distance preserving encryption* or DPE. Using the same language as for order-revealing and order-preserving encryption, we define a variant called *distance-revealing encryption* or DRE where distance is computable via some metric on the output space.

The second core operation takes a triple of points a, b and c and outputs the bit whether $d(a, c) < d(b, c)$. This type of computation is frequently used in clustering, nearest neighbor, and learning algorithms. We call a property-preserving encryption that achieves this functionality *distance-comparison revealing encryption* or DCRE. When the distance comparison on ciphertexts uses the *same* metric as on the plaintext we call *distance-comparison preserving encryption* or DCPE. The goal for this technique is to *not* allow computation of $d(a, b)$. A related notion is *approximate* distance-preserving encryption, where $\delta_1 \cdot d(a, b) \leq d(\mathsf{Enc}(a), \mathsf{Enc}(b)) \leq \delta_2 \cdot d(a, b)$ for parameters δ_1, δ_2. This can be used much like distance-preserving encryption in algorithms to give approximate guarantees.

Goals. The security achievable by DPE and DCPE is unclear even for the ideal objects. Our research on encryption for proximity consists of three tasks:

- Understanding security of DPE and constructing DPE and DRE schemes. (We expect the ideal security to vary widely based on the underlying metric properties.)
- Understanding the security of DCPE, and constructing DCPE and DCRE schemes.
- Exploring the connection to approximate distance-revealing encryption. (We believe that approximate distance-revealing encryption may be equivalent to distance-comparison preserving encryption.)

2.1 DPE and DRE

Prior work. Generalizing the work of and Li et al. [LWW+10, WMT+13], Boldyreva and Chenette [BC14] defined a *closeness-preserving tagging function* where $\mathsf{Enc}(a)$ and $\mathsf{Enc}(b)$ have associated tag sets. If the tag sets intersect, this implies that $d(a, b) \in \{equal, close\}$ with high probability; otherwise the $d(a, b)$ is deemed to be far. The construction creates a tag for every possible neighbor and thus *does not scale* to metrics where many values are considered close. Boldyreva and Chenette also present a negative result: there is a closeness function which *requires* storage proportional to the number of close neighbors [BC14, Theorem 5.2]. This result is not known to hold if the closeness function is a (well-known) metric.

Alternatively, the plaintext a can be encrypted (without tags) and the client can search for the disjunction of all neighbors of b (rather than just b). Woodage et al. apply this approach in the context of password au-

thentication with typos [WCD+17]. Both of these approaches *require time linear in the number of neighbors* and are not viable for high dimensional data.

Lastly, the co-PI [ABC+16] constructed an object called a pseudoentropic isometric which is variant of distance-preserving encryption. This definition allows that $d(\mathsf{Enc}(a), \mathsf{Enc}(b)) < d(a, b)$. It is possible to modify the construction of [ABC+16] so that $d(\mathsf{Enc}(a), \mathsf{Enc}(b)) = d(a, b)$, but this construction *only works for the set-difference metric over large alphabets and requires strong cryptographic assumptions*.

2.1.1 Proposed Work

Understanding and constructing distance-preserving encryption. There is a scattering of work on variants of distance-revealing encryption, but there is not a *solid theoretic foundation* for the object. Previous work defines noisy searchable encryption schemes but never explicitly defines distance-revealing encryption. Given the history of order-preserving encryption it is crucial to understand the security guarantees of distance-preserving and distance-revealing encryption.

The PIs first propose to study distance-revealing encryption for the Hamming metric space. Consider the Hamming metric over $\mathcal{M}^\ell$ defined as the number of positions in which two length-ℓ strings differ. Ongoing work by the PIs indicates that there is no secure distance-preserving encryption for this metric unless the size of the character set $|\mathcal{M}|$ is super-polynomial.

All distance-preserving encryption schemes are of the form $\mathsf{Enc}(x) = f(g_1(x_1), ..., g_\ell(x_\ell))$, where f is a permutation of the input coordinates and each g_i is a permutation of $\mathcal{M}$. These are the only operations that preserve distance across the entire metric space, which limits the potential security unless $\mathcal{M}$ is very large.

Whenever $|\mathcal{M}|$ is polynomial in the security parameter, the adversary can completely learn $\mathsf{Enc}(x)$ with a polynomial number of plaintext/ciphertext pairs (linear for binary strings). Prior work of the PI only provided security when $\mathcal{M}$ was superpolynomial in size [ABC+16], which our ongoing work shows is necessary. However, the PIs also plan to investigate security of distance-preserving encryption for the Hamming metric when the adversary does not know any plaintext-ciphertext pairs (i.e., a so-called ciphertext-only attack) or a very small number of them, as was previously done for order-preserving encryption [BCO11].

The PIs propose to study other metrics, in particular Euclidean, cosine-similarity, and Jaccard distance. The PIs also plan to address the case of *course grained* distance, e.g., where distance is in the set $\{far, equal, close\}$. This problem is connected to property-preserving encryption for *graph data*. A course metric can be represented by a graph where neighboring vertices are near according to the metric. Intuitively, our goal is to encrypt a graph as another graph with a subgraph of the same structure, which could lead to interesting questions in graph theory.

Understanding ideal distance-revealing encryption. In the case of *ideal* distance-revealing encryption, a foundational question is whether it can be constructed from multilinear or bilinear maps. There also may be interesting "intermediate" leakage profiles that are not ideal but leak less information than distance preserving encryption. The PI has recently used this approach to find positive results for order-revealing encryption [CLOZ16].

Distance-revealing encryption based on fuzzy extractors. We will construct distance-revealing encryption based on fuzzy extractors which are a well known primitive for deriving a stable key from a noisy source [DRS04]. They consist of a pair of algorithms: $\mathsf{Gen}(a)$, which produces a stable key key and a public value p, and $\mathsf{Rep}(b, p)$, which outputs key if $d(a, b) \in \{equal, close\}$. In ongoing work, we are constructing noisy searchable encryption from fuzzy extractors and distance-preserving encryption. Let

$\{a_i | 1 \leq i \leq n\}$ be the set of plaintexts to be stored in the database. Let Enc be a distance-preserving encryption and let H be a hash function. Rather than directly storing $\mathsf{Enc}(a_i)$ the client does the following:

- Compute $c_i \leftarrow \mathsf{Enc}(a_i)$ and then $k_i, p_i \leftarrow \mathsf{Gen}(c_i)$. Send $p_i, H(k_i)$ to server.

To search for points close to b, the client encrypts $\mathsf{Enc}(b)$ and presents this to the server which can rerun the fuzzy extractor for all stored terms. The server can then return these terms to the client. This approach has considerably less leakage than distance-preserving encryption as the server can only "compare" ciphertexts that are used in search. The stored ciphertexts are not "useful." We will extend this concept to remove the asymmetry between query and stored ciphertexts, transforming the construction into distance-revealing encryption. We expect the output of this component to be 1) evidence on which metrics are suitable for DPE and 2) constructions and analysis of DRE including constructions that use DPE.

2.2 Distance Comparison Revealing Encryption

We call encryption that supports distance comparison *distance-comparison revealing* encryption (DCRE). As described above, many learning algorithms do not require direct computation of $d(a, b)$. Rather it suffices to indicate which of two points is closer to a target point. That is, it is sufficient to compute $d(a, c) \stackrel{?}{<} d(b, c)$. Following the literature on order-revealing vs. order-preserving encryption, we call the special case where ciphertexts themselves are spatial points *distance-comparison preserving encryption* (DCPE). The hope is that the weaker functionality of DCRE and DCPE can be secure for more metrics than distance-revealing and distance-preserving encryption respectively. This leads to the following questions:

1. Can we design efficient DCPE? What security can be achieved by such schemes?
2. Can we design efficient DCRE with better security?

To answer the first question, in ongoing work we have found that distance comparison preserving functions do not seem to have a nice "recursive" property as in the case of order-preserving functions, which was crucially exploited by [BCLO09]. However, based on computer experiments, we conjecture that *distance-comparison preserving functions are approximately distance-preserving*. So far, we have proven this conjecture in one dimension. The intuition is that as the number of points in the metric spaces increases, the number of degrees of freedom decreases. The intuition is as follows:

1. Suppose that $d(b, c) = k$ is known by the attacker,
2. Learning that $d(a, c) < d(b, c)$ tells the attacker that $d(a, c) < k$.
3. Suppose the attacker also knows that $d(f, c) = k/2$ and that $d(a, c) > d(f, c)$.
4. The attacker can determine that $k/2 < d(a, c) < k$.
5. As more of these constraints are added, $d(a, c)$ is limited to smaller ranges.

That is, the encryption mechanism reveals more accurate estimations on the distance between $d(a, c)$. If this conjecture is true, then for the first question we could equivalently turn our attention to the design and analysis of an *approximately distance-preserving* encryption scheme explored next. That is, we expect a major outcome of this component to be 1) an understanding of the relationship between distance-comparison preserving encryption and approximate distance-preserving encryption (which we believe are equivalent), and 2) distance-comparison *revealing* schemes (at least based on bilinear maps, hopefully even PRFs).

2.3 Approximate Distance Revealing Encryption

Prior work. The recent GRECS work allows minimum distance between any pair of points [MKNK15]. Rather than storing or computing distance between pairs of points a, b, the work uses a primitive called a sketch-based oracle.[1] A logarithmic number of reference points $r_1, ..., r_k$ are selected and the distance between r_i and every point in the graph is computed and encrypted. This structure is transmitted to the server. Then at query time the client encrypts their pair a, b. The server finds all reference points which are connected to a, b and returns $\min_{r_i, r_j} d(a, r_i) + d(r_i, r_j) + d(r_j, b)$. Given a careful selection of the reference points this distance can be shown to approximate the minimum distance between $d(a, b)$.

A second line of work uses locality-sensitive hashes which are designed to allow more efficient computation of nearest neighbor in high dimensional spaces [DIIM04, SC08]. A locality sensitive hash is function that is more likely to have collisions when two inputs are "close" in the input space.

Kuzu, Islam and Kantarcioglu [KIK12] use locality sensitive hashing to create an approximate distance-preserving data structure. To create the index, for plaintext a the client samples multiple locality sensitive hashes $h_1(a), h_2(a), ..., h_k(a)$ and associates each output as a keyword with a using deterministic encryption. Then to query for b the client queries computes $h_1(b), ..., h_k(b)$. They then ask the server for all results that match a single hash.[2] The client locally retrieves results and then restricts to those documents that have a high number of hash matches. With good probability, these will correspond to those records that were close to the original query. Bringer et al. [BCK11, BCK09] use similar techniques but insert the output of the locality sensitive hash into a Bloom filter. Both of these works provide relatively weak security and make no attempt to hide records that match a small number of hash function outputs.

Proposed work. Distance-preserving functions are easy to characterize geometrically, in terms of a scaling factor plus flips, rotations and reflections. To approximately preserve distance, we can also "perturb" each image point within a ball of given radius. We can show that independent random such perturbations yields a function that, while not strictly DCPE, is *approximately* so, and that encrypting via an approximately DCPE function still guarantees accuracy of nearest neighbor search within the approximation. Moreover, as such perturbations can easily be derandomized, this gives an efficient *approximate* DCPE scheme from PRFs. Finally, we will conduct a separate analysis in the spirit of [BCO11] to answer the question of what privacy such a scheme provides.

A related question we plan to investigate is the privacy achievable using locality-sensitive hashing to perform nearest neighbor algorithms. To our knowledge, prior work has not explicitly defined privacy properties for locality-sensitive hashing. Moreover, the scheme of Kuzu et al. [KIK12] does not approach ideal security as the server learns which records match each subfeature. In ongoing work, we are combining locality-sensitive hashing with the recent *sample-then-hash* fuzzy extractor construction of the co-PI [CFP+16]. The idea is as follows:

1. For input a, compute $a_1 = h_1(a), a_2 = h_2(a), ..., a_k = h_k(a)$.
2. For $i = 1, ..., \ell$, sample $i_1, ..., i_\eta \xleftarrow{\$} [k]$, compute $\alpha_i = H(a_{i_1}||...||a_{i_\eta})$, and append α_i as keyword to a (using deterministic encryption).

This approach appends multiple locality sensitive hashes that individually have a small probability of matching but overall there is a high probability of a single match. The probability of finding any matches between a, b that are not close is small (see Canetti et al. [CFP+16]).

[1] This sketch-based oracle is different from the notion of a secure sketch used in fuzzy extractor constructions.

[2] To achieve this, they build an inverted index for each locality sensitive hash that allows them to retrieve the document identifiers.

3 Allowing Interaction

The direct use of property-preserving encryption has a mixed history with leakage attacks showing that deterministic and order-preserving encryption reveal the entire stored dataset for many applications (see work of the co-PI for an overview of leakage attacks [FVY$^+$17]).

Despite these attacks, the ease and speed of configuring and using PPE without replacing the entire software stack has encouraged its momentum in the commercial sector. A natural question is, *can we maintain the benefits of PPE while avoiding these attacks?*

In this section, we investigate solutions that keep intact the design principal of PPE: the only place that the database should have to change is the comparison operator (equality, comparison, or distance). The database should still be able use standard indexing mechanisms. We note it is possible to override this operator to be interactive and require help from a client without altering the overall indexing structure.

Prior work. Two main lines of work follow the approach of PPE with added interactivity. Two co-PIs recently introduced a new cryptographic approach, called POPE, to support range queries over encrypted data with stronger security than OPE [RACY16]. The server builds a novel indexing structure called a POPE tree, in which each node has a *unsorted buffer* and a sorted list of elements. Thanks to this, the scheme can perform *lazy indexing, by sorting values only when necessary*. The other work is the Arx protocol [PBP16], which builds an index for answering range queries without revealing all order relationships to the server. Encrypted values are stored in a binary tree, which is traversed privately using Yao's garbled circuits.

Another line of work that achieves a private DB solution with interaction is BlindSeer [PKV$^+$14, FVK$^+$15]. However, BlindSeer has a slightly different threat model. In particular, the system has three main players: the *server* S, *index server* IS, and *client* C. The server S, holds the DB and outsources an encrypted copy of the DB to a third party, the index server IS. The server S also builds an encrypted Bloom filter (BF) search tree over the DB and sends it to IS. The client C sends search queries to IS and obtains encrypted results, the decryption key for which is secret-shared between C and IS. These secret shares are arranged in an offline setup stage by the server S, using shuffling and homomorphic public-key encryption.

3.1 Extending and Improving POPE

Our main approach is developing PPE that operates using just-in-time advice from the client. In POPE, this took the form of asking the client to sort a small number of nodes to build out a tree and comparing ciphertexts only with those nodes. This limited leakage to comparisons between the dataset and these nodes, reducing the leakage from a quadratic number of comparisons to only linear.

Similarity of high-dimensional data. Our first task will be extending the POPE paradigm to high dimensional data. The approach uses random hyperplanes [DF08, Cha02] to form a partially-sorted search tree over multidimensional keys. The idea is as follows:

1. The server initially stores an unsorted buffer of the entire dataset.
2. The client searches for items close to a, initiating a partial sort of the buffer into a POPE tree.
3. The client generates a random hyperplane x and splits the stored elements b_i based on whether b_i is above x or not. (In the Euclidean space, this means checking whether $\langle x^T, b_i \rangle$ is positive.)
4. The client and server repeat the process with the relevant subtree until reaching an upper bound on the size of an unsorted leaf node. This leaf node which would contain a is returned to the client.

The problem with this preliminary approach is a substantial chance that nearby items will end up in different subtrees and therefore be missed in the returned set. To deal with this problem, we propose to have the client can select a *collection* of hyperplanes $x_1, ..., x_k$ and split the tree based on $\sum_{i=1}^{k} \mathsf{Sign}(\langle x_i^T, b \rangle)$. This technique allows us to control the probability that a and b will be denoted as far (lying in different subtrees) when they are close (using tail bounds for the binomial distribution).

Edit distance using partial suffix-tree encryption. Our second task is to build proximity search for edit distance. For this approach, we will build a partial version of a suffix tree which is often used in string algorithms [McC76]. Prior work by Chase and Shen [CS15] used an an encrypted suffix tree to answer substring queries.

Our idea is to apply the online algorithm of a suffix-tree construction by Ukkonen [Ukk95] and to build the suffix tree just-in-time as in the POPE protocol. Again, the advantage will be that leakage is limited to a "need-to-know" basis relative to the queries performed, rather than leaking information about the entire dataset up-front.

The client will build a single level of the suffix tree. When the client searches for strings that are close a they will traverse the partially constructed suffix tree with the client/server interactively building out the tree as necessary. This approach will require augmentations to a traditional suffix tree as the original searchable string must be stored and queryable to build the tree on demand. Balancing the speed and privacy of querying the original string represents an important tradeoff in this approach.

Improved security. Leakage attacks are particularly problematic when the adversary is able to correlate leakage from multiple queries. Forward security of searchable encryption can decouple the adversary's leakage and force them to execute their attack with less information.

POPE natively has forward security (i.e., updating an element doesn't leak information about the other elements), although it was not considered in the publication [RACY16]. This is because the scheme doesn't maintain any other index structure except the POPE tree. So, to update an element, one can simply delete the element from the node it belongs and insert an updated element to the unsorted buffer of the root.

In this proposal, we plan to investigate whether we can further reduce the leakage of the POPE scheme. In POPE, each search query leaks the ordering for the following reasons:

- The tree is a search tree. For example, all elements in the left sub-tree are smaller than every element in the right subtree.
- Once a ciphertext is inserting into a POPE tree, it never changes. This *deterministic* nature allows the attacker to trace when the ciphertext came in the tree, and how it was brought down to some leaf node.

To address the first issue, for each POPE node, the order of the links to its children may be shuffled. Since POPE is an interactive protocol, when a tree node is created, we can slightly modify the original protocol so that the client additionally change the order of the links with a randomly selected permutation.[3] Search can still be performed correctly even with this modification, since the interactive guidance of the client can help the server traverse the tree nodes correctly.

To address the second issue, we observe that when the ciphertexts in an unsorted buffer are streamed down to lower-level buffers during a search query, the client first *decrypts* them in order to indicate the correct unsorted lower-level buffer to which the ciphertexts should move. This procedure can be easily

[3]Similar ideas were used in the protocol of Ishai et al. [IKLO16] who use MPC and private information retrieval to hide tree traversal.

augmented so that the client can stream the re-reandomized cipehrtexts (instead of using the original ones deterministically) to lower-level unsorted buffers. This way, along with the shuffling idea above, we can significantly hide information about the location of lower-level buffers to which ciphertexts have been streamed down.

Another possible direction is taking advantage of ORAM. Although ORAMs are too slow to used throughout the entire system containing a large amount of data, we hope that ORAMs can be used effectively for achieving stronger privacy for the sensitive sub-part of the system (e.g., the bottom parts of the POPE tree). In fact, ORAMs have been used to minimize the leakage for SSE (symmetric searchable encryption) schemes which support keyword search over encrypted data [SPS14, GMP16, IKLO16]. We will investigate how to incorporate ORAM into POPE trees so that the resulting system enjoys stronger privacy with only marginal performance degradation.

3.2 BlindSeer in the Cloud Setting

Simplified threat model. When a database is outsourced, the database owner is often the client itself, in which case we don't need to worry about whether the plaintext data would be leaked to the client. Considering this setting will allow numerous performance improvements to Blind Seer:

- *Much simpler, more efficient setup.* In this new simpler threat model, the client can be regarded as playing as both S and C in the original BlindSeer architecture. Therefore, the client can just hold one symmetric key in contrast to the original BlindSeer system where a slow public-key encryption scheme and a random shuffling must be introduced in the setup stage. This also improves backward compatibility.
- *MPC is not necessary.* In the original BlindSeer system, in order to hide from C the BF data stored in each node of the search tree, C and IS have to execute costly MPC computation. In our setting, the client can simply ask for the necessary encrypted BF data and decrypt them, since we don't need to hide anything from the client.
- *Dynamic record insertion.* The original BlindSeer required all BF data to be known during the setup stage in order to arrange for decryption keys. Furthermore, data was encrypted with a very simple mechanism of one-time pad so that MPC computation may be reasonably efficient. Both of these choices prevent adding or removing data after initialization. In our setting, however, we don't need any key setup, nor MPC computation. Therefore, we can add BF data directly to the search tree with much more efficiency.

For use cases where this new setting is appropriate, we believe this simplified construction will be an *order of magnitude faster* than the current BlindSeer system.

Proximity of high-dimensional data. BlindSeer already provides conjunctions and range queries. We will use these queries in combination to answer proximity queries on a Euclidean space. In particular, a query that searches for the points close to (x, y) can be defined as a 2D rectangle defined by the top-left point (x_1, y_1) and the bottom-right point (y_1, y_2) can be described with range queries and conjunctions as follows:

$$x_1 \leq x \leq x_2 \ \mathsf{AND} \ \ y_1 \leq y \leq y_2.$$

This technique can be easily extended to data with multiple dimensions.

Graph structure and the shortest path. Using Bloom filters we can encode a graph structure as follows:

- For each edge (a, b) from vertex a to vertex b, we insert an encryption $c = \mathsf{Enc}((a, b)\|w(a, b))$, where $w(a, b)$ is the weight of the edge (a, b), indexed by a BF keyword 'edge:a*'.

We can find all the neighbors of vertex a along with the weight of the associated edge. Therefore, we can run variants of Dijkstra's algorithm and compute the shortest path, given two vertices a and b and a limit on the maximum number of arcs. The key to this approach is the "native" ability of Bloom filters to handle conjunction queries.

Enhancing security. In BlindSeer, the search query is executed by the client traversing the Bloom filter search tree. In particular, in each node of the tree, the client and the server execute the following:

1. For each keyword α in the query, the client computes a hash on α, based on which the look-up positions $Pos(\alpha) = (i_1, \ldots, i_\eta)$ for the BF contents are identified. Here, η is a system parameter.
2. The client sends $\{Pos(\alpha_1), \ldots, Pos(\alpha_q)\}$, where $\alpha_1, \ldots, \alpha_q$ are the keywords used in the query.
3. The client and the server execute a protocol using the look-up positions and the encrypted BF as input, so that the client knows whether the given query is satisfied.

The look-up positions that are sent from the client to the server leak $Pos(\alpha)$ to a significant degree, since q is generally small. Moreover, $Pos(\alpha)$ is a deterministic function on α. This implies that the server can infer with a reasonable probability whether two queries contain the same keyword. Given auxiliary information about query statistics, the server may be able to infer the plaintext keywords.

It is costly to reduce the leakage in the original BlindSeer system. In particular, the above step 3 (i.e., the protocol execution to check whether the given query is satisfied) is performed using secure two-party computation based on Yao's garbled circuit. To reduce the leakage, it is natural to use more sophisticated cryptographic mechanisms, but then the step 3 of secure computation would be significantly slower.

However, as noted above, we don't need secure computation in our simplified system. In this case, we can actually remove the leakage by completely hiding the look-up positions. As one promising direction, we can achieve this goal by storing the encrypted BF using ORAM techniques. Note that the look-up positions can be regarded as the metadata (i.e., "addresses") for the encrypted data (i.e., BF contents), which fits perfectly into the use case for ORAM. We expect that the resulting system will be faster than the current one, since two-party computation is known to be slower than ORAM.

3.3 Hybrid approach to improve accuracy in approximate DRE

Even after our improvements above, the fully-interactive POPE and BlindSeer schemes may still not be sufficiently scalable for the largest databases. We now propose a novel, *hybrid approach* that uses interaction alongside the Approximate DRE schemes discussed in Section 2.3:

- User encrypts keys using a noisy Approximate DRE scheme and stores them on the server
- User queries for all points within distance d of a by sending an (approximate distance-revealing) encryption a' of a to the server, along with a scaling d' of d according to the encryption scheme.
- The server finds a set S of all ciphertexts within d' of a' non-interactively
- The server enters into an interactive protocol with the user to discover and return the true set $S' \subseteq S$ of results within the true distance d of the true query point a.

By using the approach of our enhanced POPE protocol within smaller "bins" of ciphertexts on the server side, this can effectively improve on the performance and scalability of POPE by limiting the interactive portion of the protocol to a subset S of the entire database. At the same time, it improves on the communication complexity of Approximate DRE by limiting the final results to only those within the *actual* desired ball.

The privacy provided by this proposed approach is no worse than that provided by POPE or by Approximate DRE separately, but the performance is better than using either of those approaches alone. Alternatively, the hybrid approach could allow a *more noisy* approximate DRE scheme to be used, as the results will always be trimmed down using an efficient interactive protocol.

4 Analysis and Comparison

In this section of the proposal, we outline our efforts to analyze, implement, and compare the most promising schemes from the previous two sections. These efforts will be *concurrent* and *ongoing* with the theoretical developments. A main goal of our approach is to create cryptographic search systems that will be deployable in the future. Thus, it is critical to understand the practical implications of different approaches.

The main goals of this thrust of our work is to put our schemes in context of existing approaches and provide clear comparisons for practitioners and researchers alike. The work of this section will also feature the most prominent engagement with student researchers, particularly undergraduate students at our respective institutions.

4.1 Leakage analysis

All of the schemes we have proposed make some compromise of leakage for performance. Unfortunately, many prior works either focus primarily on security proofs and (sometimes novel) security definitions, or make heuristic arguments for security. It is rarely clear how to compare the leaked information. This means it is not straightforward, for example, to decide what is "more secure" between different approaches. This difficulty in comparing is not only a semantic gap. Sometimes leaked information is damaging for a particular use case and innocuous for another. This ambiguity suggests the need for standard and comprehensive benchmarks.

One of PIs made progress towards fair comparisons in a recent Systemization of Knowledge [FVY$^+$17]. We will continue in this vein and use existing metrics whenever possible to place our work in a fair context within the state of the art. Two main components of this fair comparison are 1) the use of consistent naming for leakage and 2) maintaining a partially-ordering of schemes' leakage.

For this analysis to be meaningful, cryptographic designers must be cognizant of recent attacks on PPE and related schemes such as [CGPR15, KKNO16]. These attacks often depend crucially on the datasets used and the assumed prior knowledge, and we will use the same (or equivalent) attacks against our schemes to provide a meaningful comparison.

Much of the work of this proposal is in extending existing non-interactive and interactive schemes to the multi-dimensional setting via some support of distance queries. Some existing metrics can apply directly in this setting, for example the notion of *incomparable pairs* introduced in [RACY16]. Other notions related to closeness, i.e., *pairwise distance*, also make sense in a Euclidean space. Rather than develop entirely new definitions, when possible we will use existing metrics to quantify the security improvements our schemes provide.

4.2 Implementation and experimental analysis

Throughout all phases of the project, the most promising of our schemes will be implemented and tested on realistic scenarios. Indeed, we plan to include experimental components in many of our published artifacts, with links to the open source implementations developed.

Compatibility with existing database software is a key priority of our approach. We will seek to incorporate our schemes within existing open-source projects such as the SSE library Clusion (`https://github.com/encryptedsystems/Clusion`) and the cloud-based graph database software Apache Rya (`https://rya.apache.org/`).

In order to evaluate the real-world performance of our schemes, we will be careful to use realistic and relevant datasets and queries. Tools such as the automatic test-suite generator developed in the IARPA SPAR project [HH14, VPH+15] will be used to generate repeatable and realistic experiments. As necessary, we will extend these tools to support proximity databases. We expect to extend both data and query generation. As possible, these changes will be fed back into the existing open-source projects.

Our experimental evaluations will also cover the *security* of our schemes. The approaches we propose all entail some limited information leakage. We will use empirical tools to highlight the practical implications of this limited leakage, including leakage under known attacks. For example, in [RACY16], two of the co-PIs not only evaluated the number of incomparable ciphertext pairs in theory, but also tested this leakage on a publicly-available salary database. Working with the community, we will identify a small set of datasets that highlight the different leakage profiles in the literature. We expect these datasets can be used to effectively communicate the tradeoffs between different schemes.

5 Prior Accomplishments and NSF Support

Adam O'Neill: In his Ph.D. work, the PI developed the notions of deterministic encryption [BBO07, ABO07, BFO08, BFOR08, FOR12] and order-preserving encryption [BCLO09, BCO11] to help enable search on encrypted data with processing time comparable to that for unencrypted data, while providing as-strong-as-possible security guarantees subject to this constraint. The PI has also worked on instantiating random oracles [KOS10, GOR11, LOS13], aggregate signatures [BGOY07, GLOW12], deniable encryption [OPW11], chosen-ciphertext security [KMO10, DFMO14], and functional encryption [O'N10, DIJ+13, BO13]. Since joining Georgetown, he has also been working on applications of indistinguishability obfuscation [DGL+16] and on integrating cryptography with emerging applications, such as outsourced database systems using modular order-preserving encryption [MCO+15] and privacy preserving network provenance using structured encryption [ZOSZ17].

Prior support: "EAGER: Guaranteed-Secure and Searchable Genomic Data Repositories." (PI). Proposal Number 1650419. 2016 - 2017. $99,999. "Program Obfuscation: From Theory to Practice." NSF Research Experiences for Undergraduates Supplement (PI). Supplement to Award #IIP-1362046, 2014 - 2019, $8,000.

Benjamin Fuller: In his Ph.D. work, the co-PI worked on deterministic encryption with Dr. O'Neill [FOR12, FOR15]. His main focus was on cryptography with noise developing new fuzzy extractors [FMR13, FRS16, CFP+16]. Fuzzy extractors can be thought of as a special case of distance preserving encryption where only a single point is comparable. He then oversaw evaluation and implementation of encrypted search systems at MIT Lincoln Laboratory as part of the IARPA SPAR project [IAR11] including BlindSeer codeveloped by Dr. Choi. Since joining the University of Connecticut in 2016, his work has focused on driving cryptography to practice including authentication and fuzzy extractors [HFvDD17, BKFY17, ABC+16],

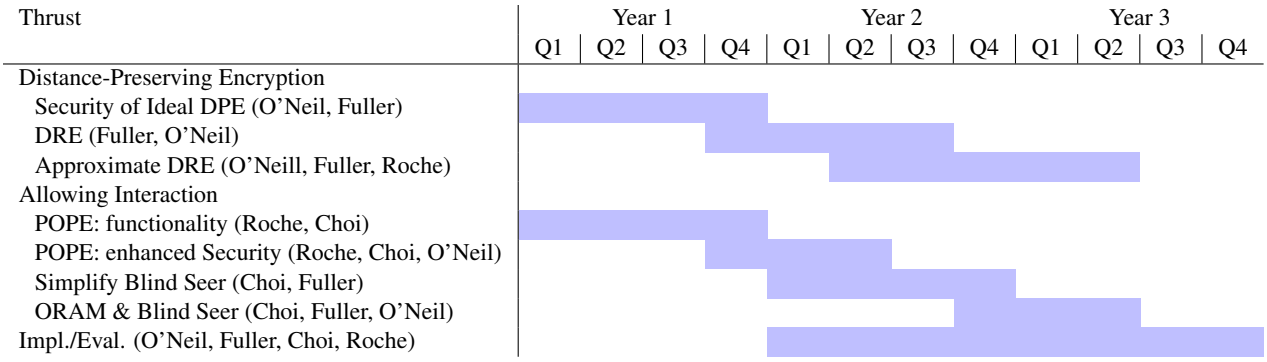

Table 1: Work schedule

secure outsourced databases [FVY+17], and multi-party computation [CFY16].

Prior NSF support: not applicable.

Seung Geol Choi: The co-PI is mainly interested in achieving privacy in practice. As for the works directly related to this proposal, he participated in the IARPA SPAR project [IAR11] as a member of the team of Columbia University and Bell Labs to build the BlindSeer system [PKV+14], and he also recently introduced a new system, called POPE, that supports range queries over encrypted data [RACY16].

He has also worked on topics related to this proposal such as ORAMs [RAC16, ACMR17, RACM17], secure multi-party computation [CEJ+07, CDMW09b, CEMY09, CHK+12, CKKZ12, CKWZ13, CKMZ14, CKS+14], and various encryption schemes [CDMW08, CDMW09a, LCL+13].

Prior NSF support: "RUI: Achieving Practical Privacy for the Cloud." (co-PI) Award number 1618269, 2016-2019, $355K.

Daniel S. Roche: This co-PI comes from an algorithms background, having worked extensively in the area of computer algebra and publishing frequently in the top venues of that area [Roc09, Roc11, GR10, GR11a, GR11b, GRT10, GRT12, HR10, AGR14, AR14, AGR15]. Recently, his interests have turned to developing improved algorithms and data structures for ensuring privacy in remote storage, which is closely related to the topic of this proposal [RAC16, RACY16, ACMR17, RACM17].

The co-PI has a proven track record of working with undergraduates and graduate students at other institutions, including multiple publications from such collaborations [AGR13, AGR14, AGR15, AR14, KRT15, GR16].

Prior NSF support: "AF: Small: RUI: Faster Arithmetic for Sparse Polynomials and Integers." (PI) Award number 1319994, 2013-2016, $123K.

Prior NSF support: "RUI: Achieving Practical Privacy for the Cloud." (co-PI) Award number 1618269, 2016-2019, $355K.

6 Schedule and Management Plan

The work described in this proposal will be performed by a combination of the PIs, graduate students at Georgetown University and University of Connecticut, and undergraduate students at all three institutions. A collaboration plan is attached as supplementary material. In this section, we provide a brief overview of the timeline of the proposed research. For each task, we have identified the relevant PI but we expect further collaboration as well. The work is summarized in Table 1.