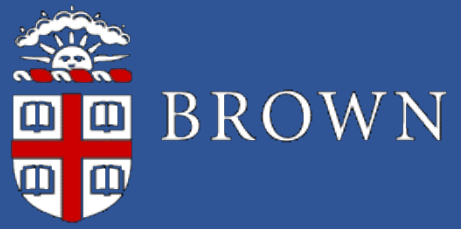
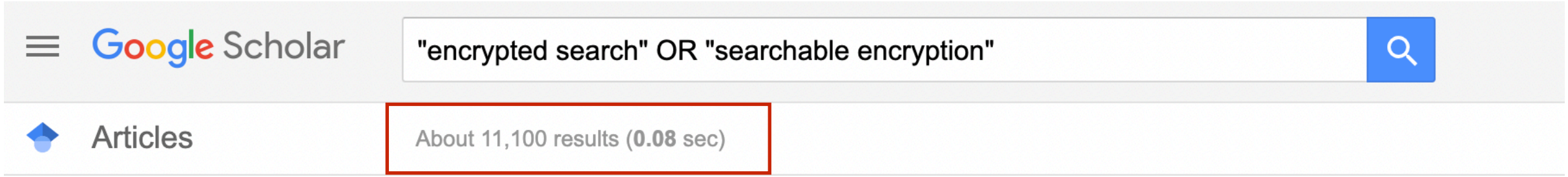
How Did We Get Here?

Seny Kamara



Encrypted Search



- Funding agencies
  - NSF
  - IARPA
  - DARPA
- Startups
  - Aroki Systems (acquired)
  - too many to list…
- Major companies
  - MongoDB, Google
  - Meta, Microsoft
  - Cisco, Hitachi, Fujitsu
  - more?


MongoDB.
Products
Solutions
Resources
Company
Pricing
Sign In
Try Free
SECURITY
Queryable Encryption.
Protect your confidential workloads.

how did we get here?

The Critics

vs.

The Dreamers

Secure Indexes*

Eu-Jin Goh
eujin@cs.stanford.edu

March 16, 2004

Abstract

A secure index is a data structure that allows a querier with a "trapdoor" for a word x to test in $O(1)$ time only if the index contains x; The index reveals no information about its contents without valid trapdoors, and trapdoors can only be generated with a secret key. Secure indexes are a natural extension of the problem of constructing data structures with privacy guarantees such as those provided by oblivious and history independent data structures. In this paper, we formally define a secure index and formulate a security model for indexes known as semantic security against adaptive chosen keyword attack (IND-CKA). We also develop an efficient IND-CKA secure index construction called Z-IDX using pseudo-random functions and Bloom filters, and show how to use Z-IDX to implement searches on encrypted data. This search scheme is the most efficient encrypted data search scheme currently known; It provides $O(1)$ search time per document, and handles compressed data, variable length words, and boolean and certain regular expression queries. The techniques developed in this paper can also be used to build encrypted searchable audit logs, private database query schemes, accumulated hashing schemes, and secure set membership tests.

1 Introduction

Keyword indexes let us search in constant time for documents containing specified keywords. Unfortunately, standard index constructions such as those using hash tables are unsuitable for indexing encrypted (and presumably sensitive) documents because they leak information about the document contents (and hence break semantic security). Informally, a secure index allows users with a "trapdoor" for a word x to test the index only for x; The index reveals no information about its contents without valid trapdoors, and trapdoors can only be generated with a secret key. Data structures with such privacy guarantees can be used to safely index the contents of semantically secure ciphertexts. We note that secure indexes do not hide information such as document size that can be obtained by simply examining the encrypted documents.

Secure indexes are a natural extension of the problem of constructing data structures with privacy guarantees such as those provided by oblivious [16] and history independent [19, 8] data structures. In oblivious (history independent) data structures, the shape (memory representation) of the data structure reveals no information about the sequence of operations applied to the data structure other than the final result. History independence is a necessary, but not sufficient, condition for a secure index; A history independent data structure guarantees nothing about the privacy of its contents, which is exactly the property required by secure indexes.

*A early version of this paper first appeared on the Cryptology ePrint Archive on October 7th 2003.

1

The Foundational Papers

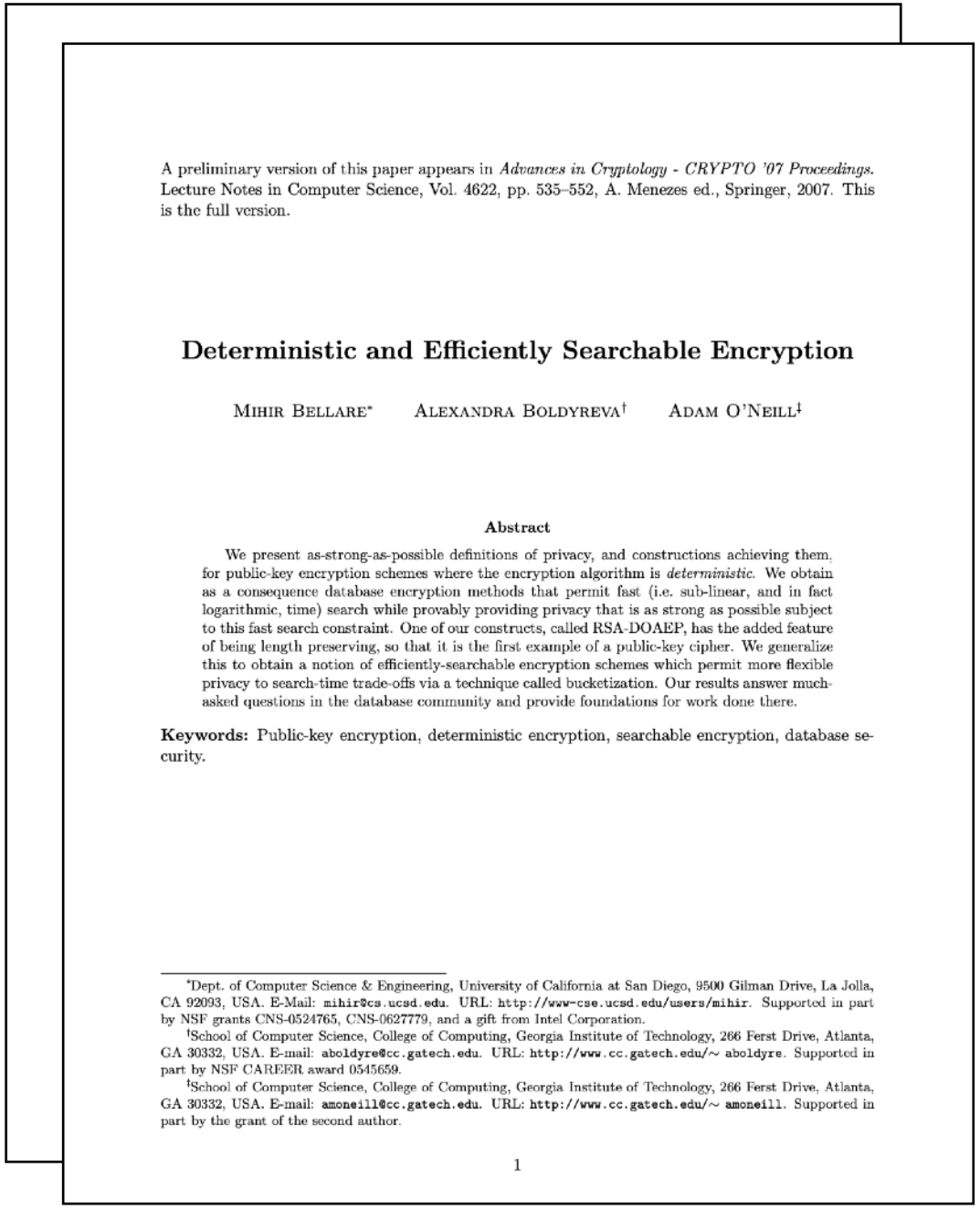
[Song-Wagner-Perrig'00]

- Keyword search on encrypted documents?
- Yes in time O(|document collection|)
- with *CPA*-security
- leakage not captured

[Chang-Mitzenmacher'05, **Goh03**]

- Yes in time O(# of documents)
- CKA-security with no query privacy
- leakage not captured

Rejected! x ∞



A preliminary version of this paper appears in *Advances in Cryptology - CRYPTO '07 Proceedings*. Lecture Notes in Computer Science, Vol. 4622, pp. 535–552, A. Menezes ed., Springer, 2007. This is the full version.

Deterministic and Efficiently Searchable Encryption

MIHIR BELLARE* ALEXANDRA BOLDYREVA† ADAM O'NEILL‡

Abstract

We present as-strong-as-possible definitions of privacy, and constructions achieving them, for public-key encryption schemes where the encryption algorithm is *deterministic*. We obtain as a consequence database encryption methods that permit fast (i.e. sub-linear, and in fact logarithmic, time) search while provably providing privacy that is as strong as possible subject to this fast search constraint. One of our constructs, called RSA-DOAEP, has the added feature of being length preserving, so that it is the first example of a public-key cipher. We generalize this to obtain a notion of efficiently-searchable encryption schemes which permit more flexible privacy to search-time trade-offs via a technique called bucketization. Our results answer much-asked questions in the database community and provide foundations for work done there.

Keywords: Public-key encryption, deterministic encryption, searchable encryption, database security.

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-Mail: mihir@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/mihir. Supported in part by NSF grants CNS-0524765, CNS-0627779, and a gift from Intel Corporation.

†School of Computer Science, College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA. E-mail: aboldyre@cc.gatech.edu. URL: http://www.cc.gatech.edu/~ aboldyre. Supported in part by NSF CAREER award 0545659.

‡School of Computer Science, College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA. E-mail: amoneill@cc.gatech.edu. URL: http://www.cc.gatech.edu/~ amoneill. Supported in part by the grant of the second author.

The Foundational Papers

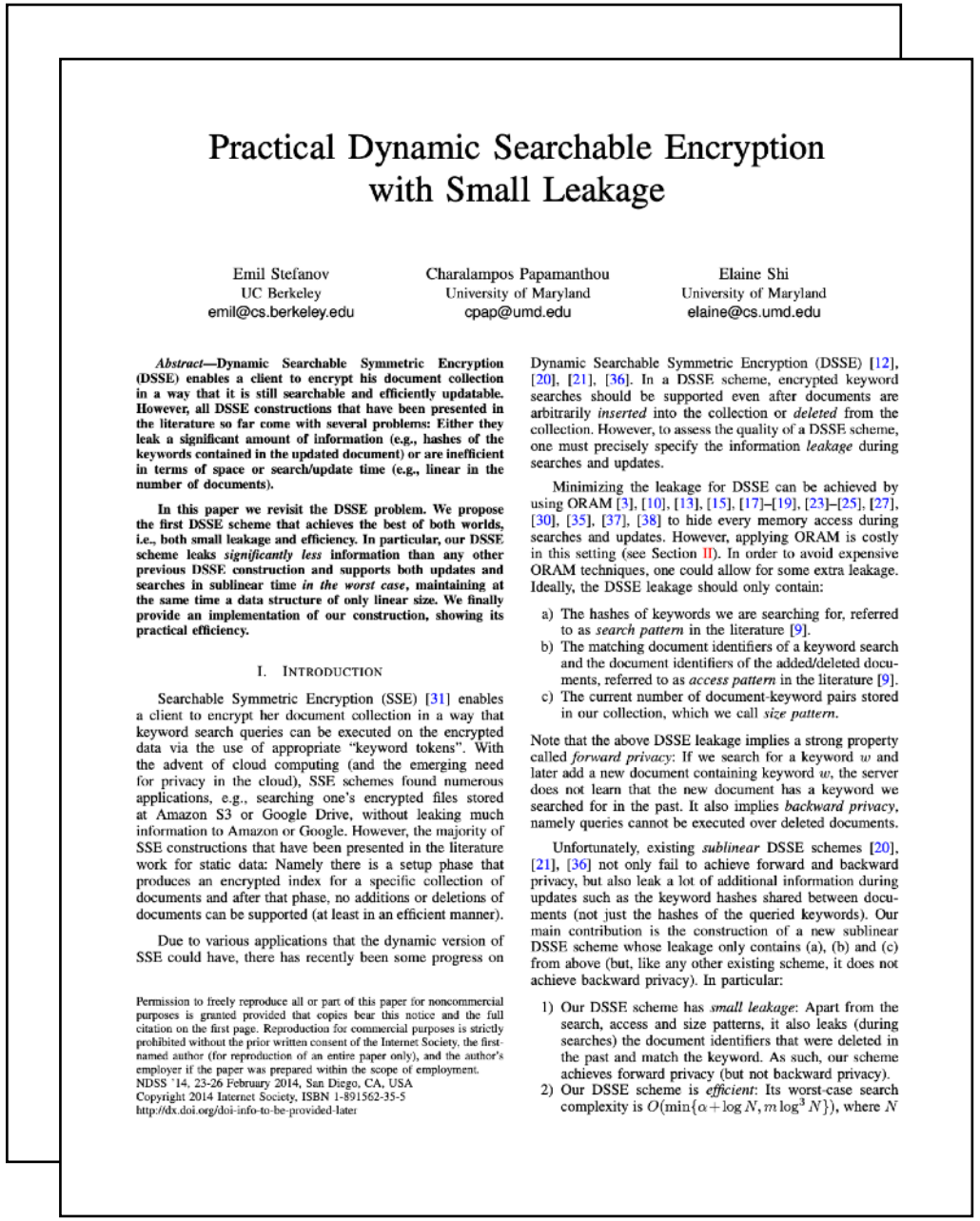
- [Curtmola-Garay-K.-Ostrovsky'06]
  - Yes in time O(OPT)
  - non-adaptive & adaptive CKA-security w/ query privacy
  - with formally defined leakage
    - no leakage vs. snapshot adv.
    - query equality & response id vs. persistent adv.

- [Bellare-Boldyreva-O'Neill'06]
  - Yes in time O(OPT)
  - Security for high-entropy data
    - frequency leakage vs. snapshot adv.
    - frequency & query eq. leakage vs. persistent adv.

C: Who cares? This can be solved with ORAM and MPC!

A: Sure but we need optimal-time search & concrete efficiency

Q: Can we handle dynamic data in O(OPT)?



Practical Dynamic Searchable Encryption with Small Leakage

Emil Stefanov
UC Berkeley
emil@cs.berkeley.edu

Charalampos Papamanthou
University of Maryland
cpap@umd.edu

Elaine Shi
University of Maryland
elaine@cs.umd.edu

***Abstract*—Dynamic Searchable Symmetric Encryption (DSSE) enables a client to encrypt his document collection in a way that it is still searchable and efficiently updatable. However, all DSSE constructions that have been presented in the literature so far come with several problems: Either they leak a significant amount of information (e.g., hashes of the keywords contained in the updated document) or are inefficient in terms of space or search/update time (e.g., linear in the number of documents).**

In this paper we revisit the DSSE problem. We propose the first DSSE scheme that achieves the best of both worlds, i.e., both small leakage and efficiency. In particular, our DSSE scheme leaks *significantly less* information than any other previous DSSE construction and supports both updates and searches in sublinear time *in the worst case*, maintaining at the same time a data structure of only linear size. We finally provide an implementation of our construction, showing its practical efficiency.

I. Introduction

Searchable Symmetric Encryption (SSE) [31] enables a client to encrypt her document collection in a way that keyword search queries can be executed on the encrypted data via the use of appropriate "keyword tokens". With the advent of cloud computing (and the emerging need for privacy in the cloud), SSE schemes found numerous applications, e.g., searching one's encrypted files stored at Amazon S3 or Google Drive, without leaking much information to Amazon or Google. However, the majority of SSE constructions that have been presented in the literature work for static data: Namely there is a setup phase that produces an encrypted index for a specific collection of documents and after that phase, no additions or deletions of documents can be supported (at least in an efficient manner).

Due to various applications that the dynamic version of SSE could have, there has recently been some progress on Dynamic Searchable Symmetric Encryption (DSSE) [12], [20], [21], [36]. In a DSSE scheme, encrypted keyword searches should be supported even after documents are arbitrarily *inserted* into the collection or *deleted* from the collection. However, to assess the quality of a DSSE scheme, one must precisely specify the information *leakage* during searches and updates.

Minimizing the leakage for DSSE can be achieved by using ORAM [3], [10], [13], [15], [17]–[19], [23]–[25], [27], [30], [35], [37], [38] to hide every memory access during searches and updates. However, applying ORAM is costly in this setting (see Section II). In order to avoid expensive ORAM techniques, one could allow for some extra leakage. Ideally, the DSSE leakage should only contain:

a) The hashes of keywords we are searching for, referred to as *search pattern* in the literature [9].
b) The matching document identifiers of a keyword search and the document identifiers of the added/deleted documents, referred to as *access pattern* in the literature [9].
c) The current number of document-keyword pairs stored in our collection, which we call *size pattern*.

Note that the above DSSE leakage implies a strong property called *forward privacy*: If we search for a keyword w and later add a new document containing keyword w, the server does not learn that the new document has a keyword we searched for in the past. It also implies *backward privacy*, namely queries cannot be executed over deleted documents.

Unfortunately, existing *sublinear* DSSE schemes [20], [21], [36] not only fail to achieve forward and backward privacy, but also leak a lot of additional information during updates such as the keyword hashes shared between documents (not just the hashes of the queried keywords). Our main contribution is the construction of a new sublinear DSSE scheme whose leakage only contains (a), (b) and (c) from above (but, like any other existing scheme, it does not achieve backward privacy). In particular:

1) Our DSSE scheme has *small leakage*: Apart from the search, access and size patterns, it also leaks (during searches) the document identifiers that were deleted in the past and match the keyword. As such, our scheme achieves forward privacy (but not backward privacy).
2) Our DSSE scheme is *efficient*: Its worst-case search complexity is $O(\min\{\alpha+\log N, m\log^3 N\})$, where N

NDSS '14, 23-26 February 2014, San Diego, CA, USA
 ISBN 1-891562-35-5
http://dx.doi.org/doi-info-to-be-provided-later

The Dynamic Papers

- [K.-Papamanthou-Roeder'12]
  - Dynamic scheme with O(OPT) queries
  - dynamic security definition
  - Leakage
    - query equality, response identity
    - update & past query correlations
- [Stefanov-Papamanthou-Shi'14]
  - Dynamic in time $O(OPT \cdot \log^3 N)$ with $O(N^{\alpha})$ client storage
  - with *forward privacy*
  - Leakage
    - query equality, response identity

Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation

David Cash*, Joseph Jaeger*, Stanislaw Jarecki†, Charanjit Jutla‡,
Hugo Krawczyk‡, Marcel-Cătălin Roşu‡, and Michael Steiner‡
*Rutgers University
†University of California, Irvine
‡IBM Research

***Abstract*—We design and implement dynamic symmetric searchable encryption schemes that efficiently and privately search server-held encrypted databases with tens of billions of record-keyword pairs. Our basic theoretical construction supports single-keyword searches and offers asymptotically optimal server index size, fully parallel searching, and minimal leakage. Our implementation effort brought to the fore several factors ignored by earlier coarse-grained theoretical performance analyses, including low-level space utilization, I/O parallelism and goodput. We accordingly introduce several optimizations to our theoretically optimal construction that model the prototype's characteristics designed to overcome these factors. All of our schemes and optimizations are proven secure and the information leaked to the untrusted server is precisely quantified. We evaluate the performance of our prototype using two very large datasets: a synthesized census database with 100 million records and hundreds of keywords per record and a multi-million webpage collection that includes Wikipedia as a subset. Moreover, we report on an implementation that uses the dynamic SSE schemes developed here as the basis for supporting recent SSE advances, including complex search queries (e.g., Boolean queries) and richer operational settings (e.g., query delegation), in the above terabyte-scale databases.**

I. Introduction

BACKGROUND. Searchable symmetric encryption (SSE) allows one to store data at an untrusted server and later search the data for records (or documents) matching a given keyword while maintaining privacy. Many recent works [3]–[5], [7], [9], [14], [15], [17], [19], [21] studied SSE and provided solutions with varying trade-offs between security, efficiency, and the ability to securely update the data after it has been encrypted and uploaded. These constructions aim at practical efficiency, in contrast to generic cryptographic tools like homomorphic encryption or multiparty computation which are highly secure but not likely to be efficient in practice.

Large data sizes motivate storage outsourcing, so to be useful an SSE scheme must scale well. Existing SSE schemes employ only symmetric cryptography operations and standard data structures and thus show potential for practical efficiency, but obstacles remain. While most constructions have theoretically optimal search times that scale only with the number of documents matching the query, the performance of their implementations on large datasets is less clear. Factors like I/O latency, storage utilization, and the variance of real-world dataset distributions degrade the practical performance of theoretically efficient SSE schemes. One critical source of inefficiency in practice (often ignored in theory) is a complete lack of locality and parallelism: To execute a search, most prior SSE schemes *sequentially* read each result from storage at a pseudorandom position, and the only known way to avoid this *while maintaining privacy* involves padding the server index to a prohibitively large size.

CONTRIBUTIONS. We give the first SSE implementation that can encrypt and search on datasets with tens of billions of record/keyword pairs. To design our scheme, we start with a new, simple, theoretical SSE construction that uses a generic dictionary structure to already achieve an asymptotic improvement over prior SSE schemes, giving optimal leakage, server size, search computation, and parallelism in search. This starting point can be seen as a generalization and simplification of the more ad-hoc techniques of [3]. We show how to make the scheme *dynamic*, meaning that the data can be changed after encryption: Our scheme can easily support additions to the data, as well as deletions via revocation lists.

Because the scheme uses a generic dictionary that itself has no security properties, it allows for several extensions and modifications with only small changes to the security proofs. In particular, our implementation effort showed that disk I/O utilization remained a bottleneck which prevented scaling; so we extend our basic construction to improve locality and throughput. These extensions preserve privacy with slightly different leakages that we analyze with formal security proofs. Below we describe the techniques behind results in more detail, starting with the new theoretical scheme that we extend later, and then compare our results to prior work.

BASIC CONSTRUCTION. Our scheme is very simple (see Figure 2): It associates with each record/keyword pair a pseudorandom label, and then for each pair stores the encrypted record identifier with that label in a generic dictionary data structure. We derive the labels so that the client, on input a keyword to query, can compute a keyword-specific short key allowing the server to search by first recomputing the labels, then retrieving the encrypted identifiers from the dic-

NDSS '14, 23-26 February 2014, San Diego, CA, USA

http://dx.doi.org/10.14722/ndss.2014.23264

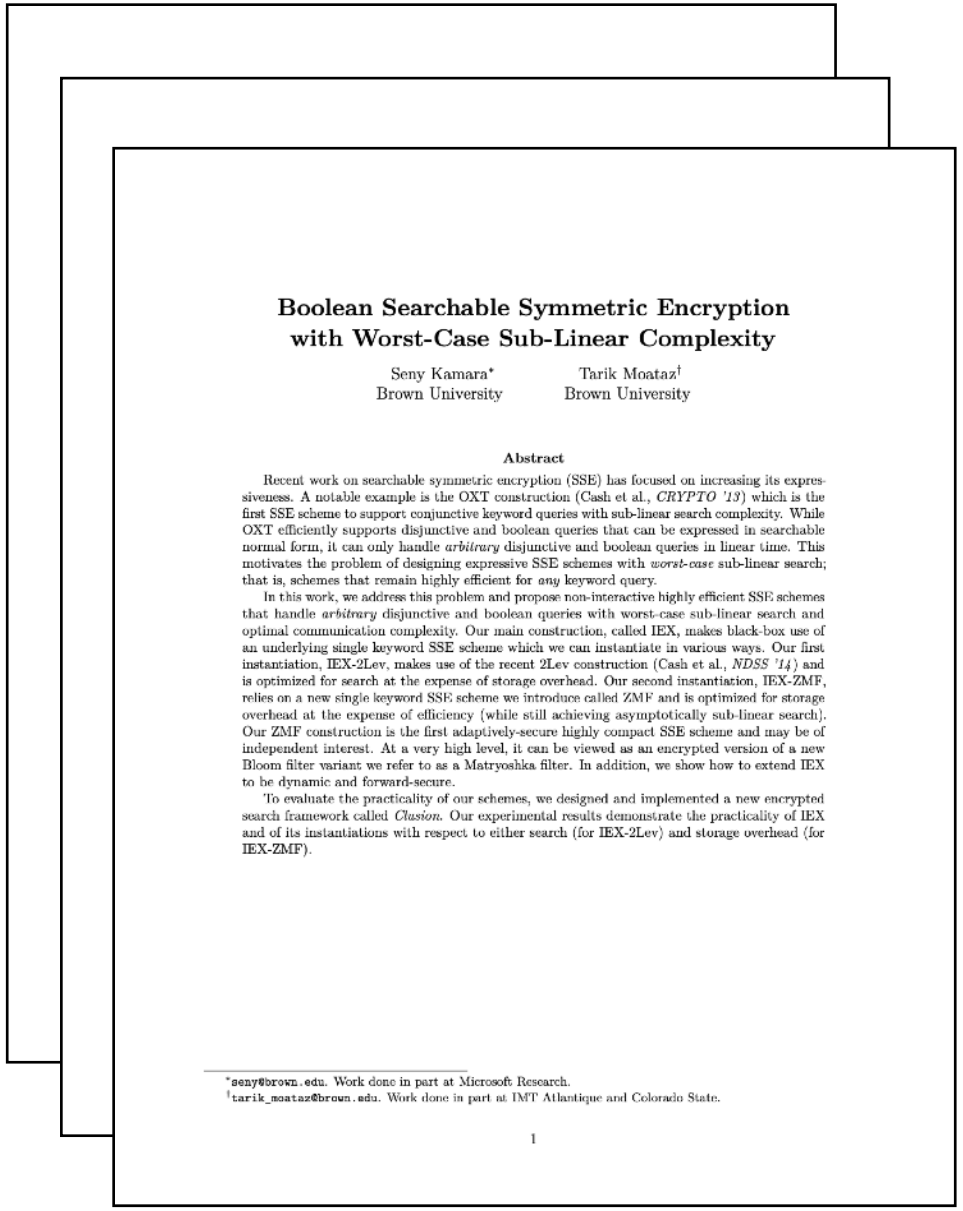
The Dynamic Papers

- [Bost'16]
  - Dyn. in $O(OPT+dels_0)$ w/ $O(W \log n)$ client storage
  - *formal definition* of forward privacy
  - forward privacy based on TDPs
  - Leakage
    - query equality, response identity
- [Cash et al.'14]
  - Dynamic in $O(OPT)$ w/ $O(W \log N)$ client storage
  - can be forward private with $O(\#resp)$ communication
  - Leakage
    - query equality, response identity

C: What's with all this leakage anyway? Just use ORAM!

A: ORAM can't be used "as-is" and leaks as well

Q: Can we handle complex queries in O(OPT)?



Boolean Searchable Symmetric Encryption with Worst-Case Sub-Linear Complexity

Seny Kamara* Brown University
Tarik Moataz† Brown University

Abstract

Recent work on searchable symmetric encryption (SSE) has focused on increasing its expressiveness. A notable example is the OXT construction (Cash et al., *CRYPTO '13*) which is the first SSE scheme to support conjunctive keyword queries with sub-linear search complexity. While OXT efficiently supports disjunctive and boolean queries that can be expressed in searchable normal form, it can only handle *arbitrary* disjunctive and boolean queries in linear time. This motivates the problem of designing expressive SSE schemes with *worst-case* sub-linear search; that is, schemes that remain highly efficient for *any* keyword query.

In this work, we address this problem and propose non-interactive highly efficient SSE schemes that handle *arbitrary* disjunctive and boolean queries with worst-case sub-linear search and optimal communication complexity. Our main construction, called IEX, makes black-box use of an underlying single keyword SSE scheme which we can instantiate in various ways. Our first instantiation, IEX-2Lev, makes use of the recent 2Lev construction (Cash et al., *NDSS '14*) and is optimized for search at the expense of storage overhead. Our second instantiation, IEX-ZMF, relies on a new single keyword SSE scheme we introduce called ZMF and is optimized for storage overhead at the expense of efficiency (while still achieving asymptotically sub-linear search). Our ZMF construction is the first adaptively-secure highly compact SSE scheme and may be of independent interest. At a very high level, it can be viewed as an encrypted version of a new Bloom filter variant we refer to as a Matryoshka filter. In addition, we show how to extend IEX to be dynamic and forward-secure.

To evaluate the practicality of our schemes, we designed and implemented a new encrypted search framework called *Clusion*. Our experimental results demonstrate the practicality of IEX and of its instantiations with respect to either search (for IEX-2Lev) and storage overhead (for IEX-ZMF).

*seny@brown.edu. Work done in part at Microsoft Research.
†tarik_moataz@brown.edu. Work done in part at IMT Atlantique and Colorado State.

1

The Boolean Papers

- [Cash et al. '13]
  - Conjunctive & SNF in o(n) w/ opt. comm. in 1 round
  - Complex leakage
- [Pappas et al.'14]
  - Boolean in o(n) w/ non-opt. comm. in multiple rounds
  - Complex leakage
- [K.-Moataz'17]
  - Boolean in o(n) w/ opt. comm. in 1 round
  - Complex leakage

Rich Queries on Encrypted Data: Beyond Exact Matches*

Sky Faber** Stanislaw Jarecki*** Hugo Krawczyk†
Quan Nguyen‡ Marcel Rosu§ Michael Steiner¶

Abstract. We extend the searchable symmetric encryption (SSE) protocol of [Cash et al., Crypto'13] adding support for range, substring, wildcard, and phrase queries, in addition to the Boolean queries supported in the original protocol. Our techniques apply to the basic single-client scenario underlying the common SSE setting as well as to the more complex Multi-Client and Outsourced Symmetric PIR extensions of [Jarecki et al., CCS'13]. We provide performance information based on our prototype implementation, showing the practicality and scalability of our techniques to very large databases, thus extending the performance results of [Cash et al., NDSS'14] to these rich and comprehensive query types.

1 Introduction

Searchable symmetric encryption (SSE) addresses a setting where a client outsources an encrypted database (or document/file collection) to a remote server $\mathcal{E}$ such that the client, which only stores a cryptographic key, can later search the collection at $\mathcal{E}$ while hiding information about the database and queries from $\mathcal{E}$. Leakage to $\mathcal{E}$ is to be confined to well-defined forms of data-access and query patterns while preventing disclosure of explicit data and query plaintext values. SSE has been extensively studied [25,12,7,10,8,18,15,17,14,6,13,5,20,19], particularly in last years due to the popularity of clouds and data outsourcing, focusing almost exclusively on single-keyword search.

Recently, Cash et al. [6] and Pappas et al. [20] presented the first SSE solutions that go well beyond single-keyword search by supporting Boolean queries on multiple keywords in sublinear time. In particular, [6,5] build a very scalable system with demonstrated practical performance with databases containing indexes in the order of tens of billions document-keyword pairs. In this work we extend the search capabilities of the system from [6] (*referred to as the OXT protocol*) by supporting range queries (e.g., return all records of people born between two given dates), substring queries (e.g., return records with textual information containing a given pattern, say 'crypt'), wildcard queries (combining substrings with one or more single-character wildcards), and phrase queries (return records that contain the phrase "searchable encryption"). Moreover, by preserving the overall system design and optimized data structures of [5], we can run any of these new queries in combination with Boolean-search capabilities (e.g., combining a range and/or substring query with a conjunction of additional keywords/ranges/substrings) and we can do so while preserving the scalability of the system and additional properties such as support for *dynamic data*.

We also show how to extend our techniques to the more involved multi-client SSE scenarios studied by Jarecki et al. [13]. In the first scenario, denoted MC-SSE, the owner of the data, $\mathcal{D}$, outsources its data to a remote server $\mathcal{E}$ in encrypted form and later allows multiple clients to access the data via search queries and according to an authorization policy managed by $\mathcal{D}$. The system is intended to limit the information learned

* Preliminary version published at ESORICS 2015 [11]
** U. California Irvine. Email: fabers@uci.edu.
*** U. California Irvine. Email: stasio@ics.uci.edu.
† IBM Research. Email: hugo@ee.technion.ac.il.
‡ Google, Inc. Email: quannguyen@google.com.
§ Bloomberg. Email: marcelrosu@gmail.com.
¶ IBM Research. Email: steiner@acm.org.

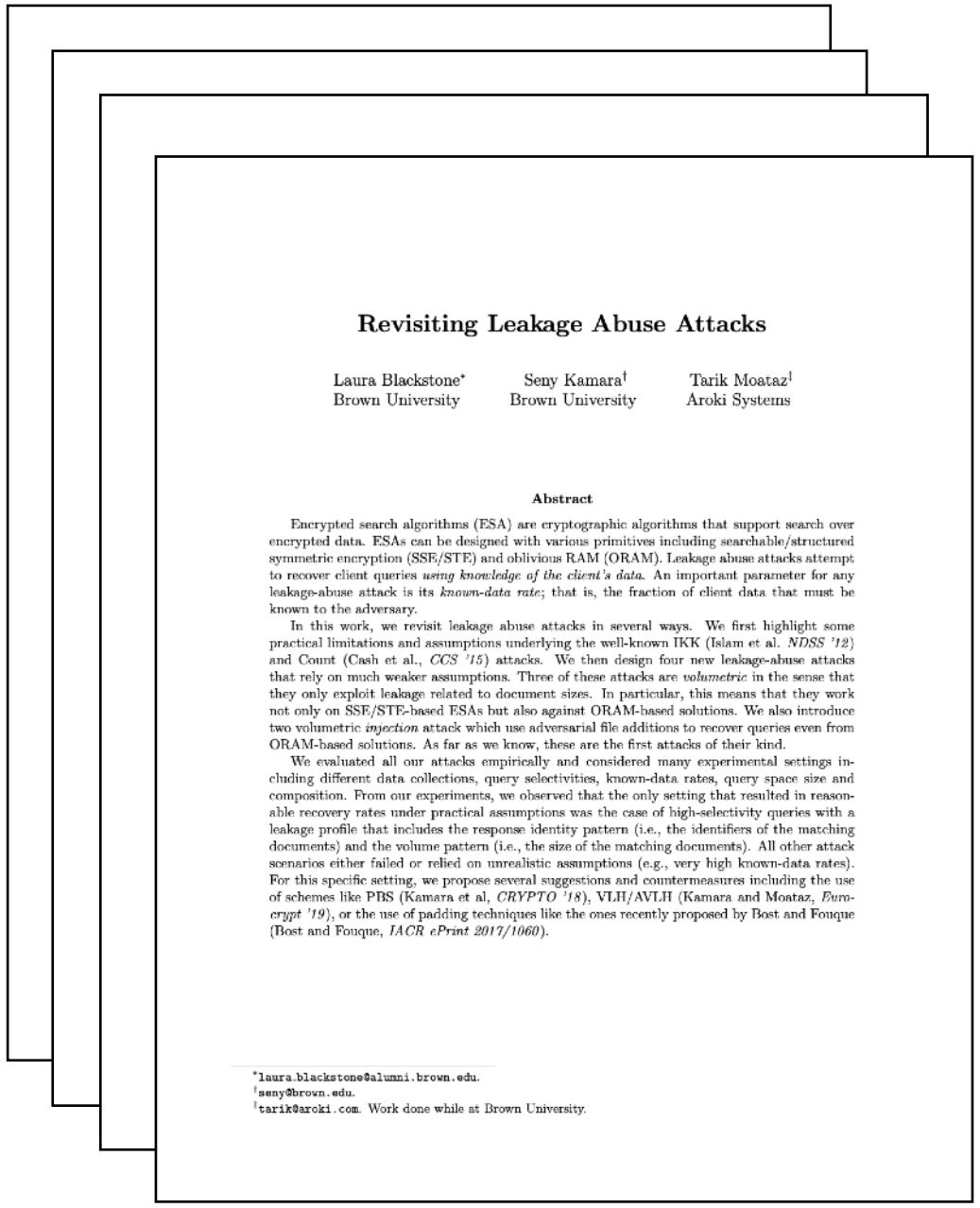
The Range Papers

- [Pappas et al.'14]
  - ranges in o(n) w/ non-opt. comm. and multiple rounds
  - Complex leakage
- [Demertzis et al.'16]
  - range in o(n) w/ opt. comm and 1 round
  - Complex leakage
- [Faber et al.'15]
  - range & substring in o(n) w/ opt. comm. and 1 round
  - Complex leakage

C: What's the point of formal leakage profiles anyway?

A: So we can be precise about leakage and attack it

Q: Can we exploit these leakage profiles in practice?



Revisiting Leakage Abuse Attacks

Laura Blackstone* Brown University
Seny Kamara† Brown University
Tarik Moataz‡ Aroki Systems

Abstract

Encrypted search algorithms (ESA) are cryptographic algorithms that support search over encrypted data. ESAs can be designed with various primitives including searchable/structured symmetric encryption (SSE/STE) and oblivious RAM (ORAM). Leakage abuse attacks attempt to recover client queries *using knowledge of the client's data*. An important parameter for any leakage-abuse attack is its *known-data rate*; that is, the fraction of client data that must be known to the adversary.

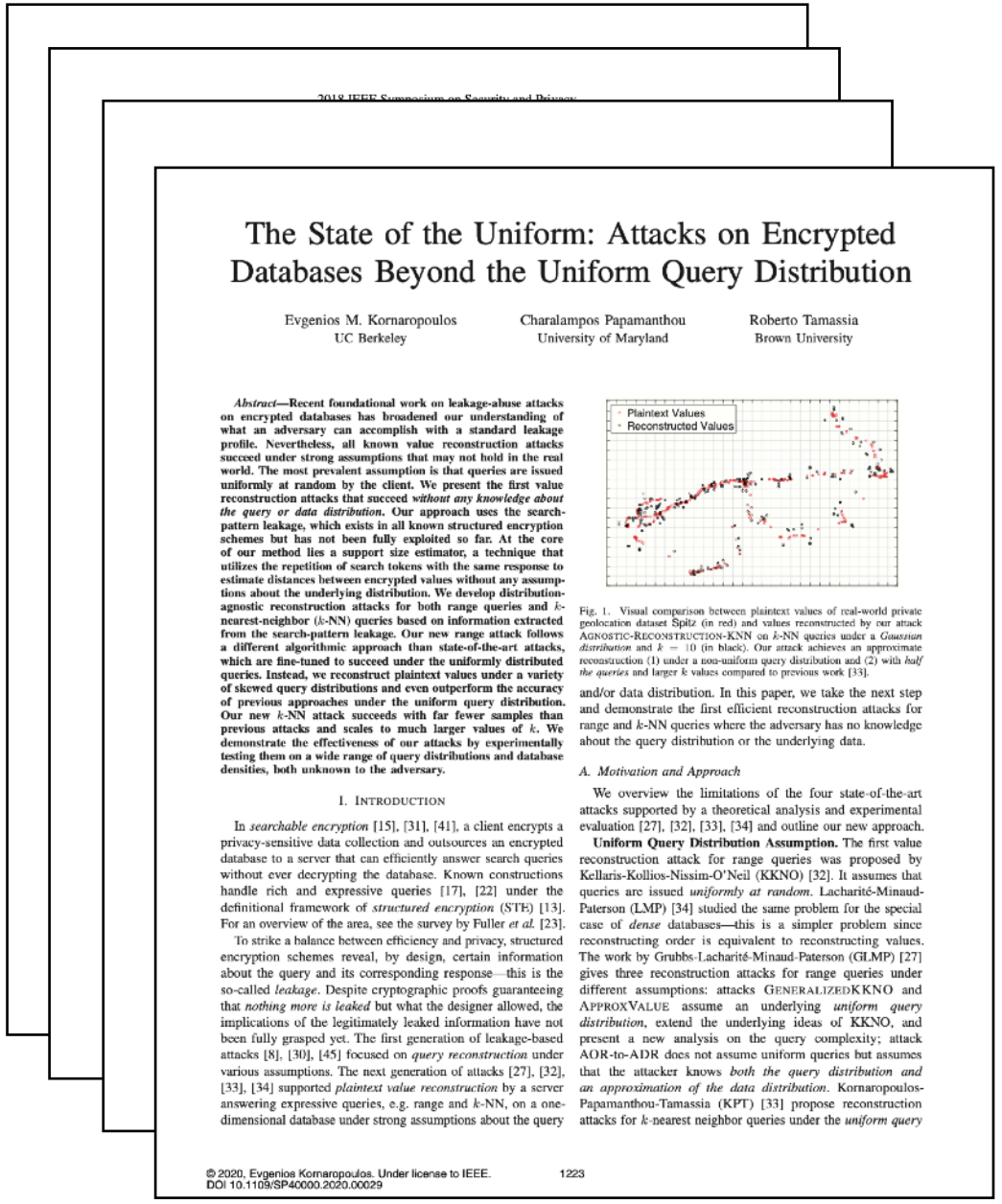
In this work, we revisit leakage abuse attacks in several ways. We first highlight some practical limitations and assumptions underlying the well-known IKK (Islam et al. *NDSS '12*) and Count (Cash et al., *CCS '15*) attacks. We then design four new leakage-abuse attacks that rely on much weaker assumptions. Three of these attacks are *volumetric* in the sense that they only exploit leakage related to document sizes. In particular, this means that they work not only on SSE/STE-based ESAs but also against ORAM-based solutions. We also introduce two volumetric *injection* attack which use adversarial file additions to recover queries even from ORAM-based solutions. As far as we know, these are the first attacks of their kind.

We evaluated all our attacks empirically and considered many experimental settings including different data collections, query selectivities, known-data rates, query space size and composition. From our experiments, we observed that the only setting that resulted in reasonable recovery rates under practical assumptions was the case of high-selectivity queries with a leakage profile that includes the response identity pattern (i.e., the identifiers of the matching documents) and the volume pattern (i.e., the size of the matching documents). All other attack scenarios either failed or relied on unrealistic assumptions (e.g., very high known-data rates). For this specific setting, we propose several suggestions and countermeasures including the use of schemes like PBS (Kamara et al, *CRYPTO '18*), VLH/AVLH (Kamara and Moataz, *Eurocrypt '19*), or the use of padding techniques like the ones recently proposed by Bost and Fouque (Bost and Fouque, *IACR ePrint 2017/1060*).

*laura.blackstone@alumni.brown.edu.
†seny@brown.edu.
‡tarik@aroki.com. Work done while at Brown University.

The Cryptanalysis Papers

- [Islam-Kuzu-Kantaciorglu'12]
  - exploits co-occurrence pattern
  - used to motivate 100s of papers but…
  - …doesn't work unless adversary knows all the data
- [Cash-Grubbs-Perry-Ristenpart'15]
  - exploits co-occurrence pattern
  - works if adversary knows almost all the data
- [Zhang-Papamanthou-Katz'16]
  - exploits response identity and ability to inject files
- [Blackstone-K.-Moataz'19]
  - exploits only volume pattern
  - applicable to ORAM-based solutions as well



The State of the Uniform: Attacks on Encrypted Databases Beyond the Uniform Query Distribution

Evgenios M. Kornaropoulos, UC Berkeley
Charalampos Papamanthou, University of Maryland
Roberto Tamassia, Brown University

***Abstract*—Recent foundational work on leakage-abuse attacks on encrypted databases has broadened our understanding of what an adversary can accomplish with a standard leakage profile. Nevertheless, all known value reconstruction attacks succeed under strong assumptions that may not hold in the real world. The most prevalent assumption is that queries are issued uniformly at random by the client. We present the first value reconstruction attacks that succeed *without any knowledge about the query or data distribution*. Our approach uses the search-pattern leakage, which exists in all known structured encryption schemes but has not been fully exploited so far. At the core of our method lies a support size estimator, a technique that utilizes the repetition of search tokens with the same response to estimate distances between encrypted values without any assumptions about the underlying distribution. We develop distribution-agnostic reconstruction attacks for both range queries and k-nearest-neighbor (k-NN) queries based on information extracted from the search-pattern leakage. Our new range attack follows a different algorithmic approach than state-of-the-art attacks, which are fine-tuned to succeed under the uniformly distributed queries. Instead, we reconstruct plaintext values under a variety of skewed query distributions and even outperform the accuracy of previous approaches under the uniform query distribution. Our new k-NN attack succeeds with far fewer samples than previous attacks and scales to much larger values of k. We demonstrate the effectiveness of our attacks by experimentally testing them on a wide range of query distributions and database densities, both unknown to the adversary.**

I. Introduction

In *searchable encryption* [15], [31], [41], a client encrypts a privacy-sensitive data collection and outsources an encrypted database to a server that can efficiently answer search queries without ever decrypting the database. Known constructions handle rich and expressive queries [17], [22] under the definitional framework of *structured encryption* (STE) [13]. For an overview of the area, see the survey by Fuller *et al.* [23].

To strike a balance between efficiency and privacy, structured encryption schemes reveal, by design, certain information about the query and its corresponding response—this is the so-called *leakage*. Despite cryptographic proofs guaranteeing that *nothing more is leaked* but what the designer allowed, the implications of the legitimately leaked information have not been fully grasped yet. The first generation of leakage-based attacks [8], [30], [45] focused on *query reconstruction* under various assumptions. The next generation of attacks [27], [32], [33], [34] supported *plaintext value reconstruction* by a server answering expressive queries, e.g. range and k-NN, on a one-dimensional database under strong assumptions about the query and/or data distribution. In this paper, we take the next step and demonstrate the first efficient reconstruction attacks for range and k-NN queries where the adversary has no knowledge about the query distribution or the underlying data.

Fig. 1. Visual comparison between plaintext values of real-world private geolocation dataset Spitz (in red) and values reconstructed by our attack AGNOSTIC-RECONSTRUCTION-KNN on k-NN queries under a *Gaussian distribution* and $k = 10$ (in black). Our attack achieves an approximate reconstruction (1) under a non-uniform query distribution and (2) with *half the queries* and larger k values compared to previous work [33].

A. Motivation and Approach

We overview the limitations of the four state-of-the-art attacks supported by a theoretical analysis and experimental evaluation [27], [32], [33], [34] and outline our new approach.

Uniform Query Distribution Assumption. The first value reconstruction attack for range queries was proposed by Kellaris-Kollios-Nissim-O'Neil (KKNO) [32]. It assumes that queries are issued *uniformly at random*. Lacharité-Minaud-Paterson (LMP) [34] studied the same problem for the special case of *dense* databases—this is a simpler problem since reconstructing order is equivalent to reconstructing values. The work by Grubbs-Lacharité-Minaud-Paterson (GLMP) [27] gives three reconstruction attacks for range queries under different assumptions: attacks GENERALIZEDKKNO and APPROXVALUE assume an underlying *uniform query distribution*, extend the underlying ideas of KKNO, and present a new analysis on the query complexity; attack AOR-to-ADR does not assume uniform queries but assumes that the attacker knows *both the query distribution and an approximation of the data distribution*. Kornaropoulos-Papamanthou-Tamassia (KPT) [33] propose reconstruction attacks for k-nearest neighbor queries under the *uniform query*

DOI 10.1109/SP40000.2020.00029

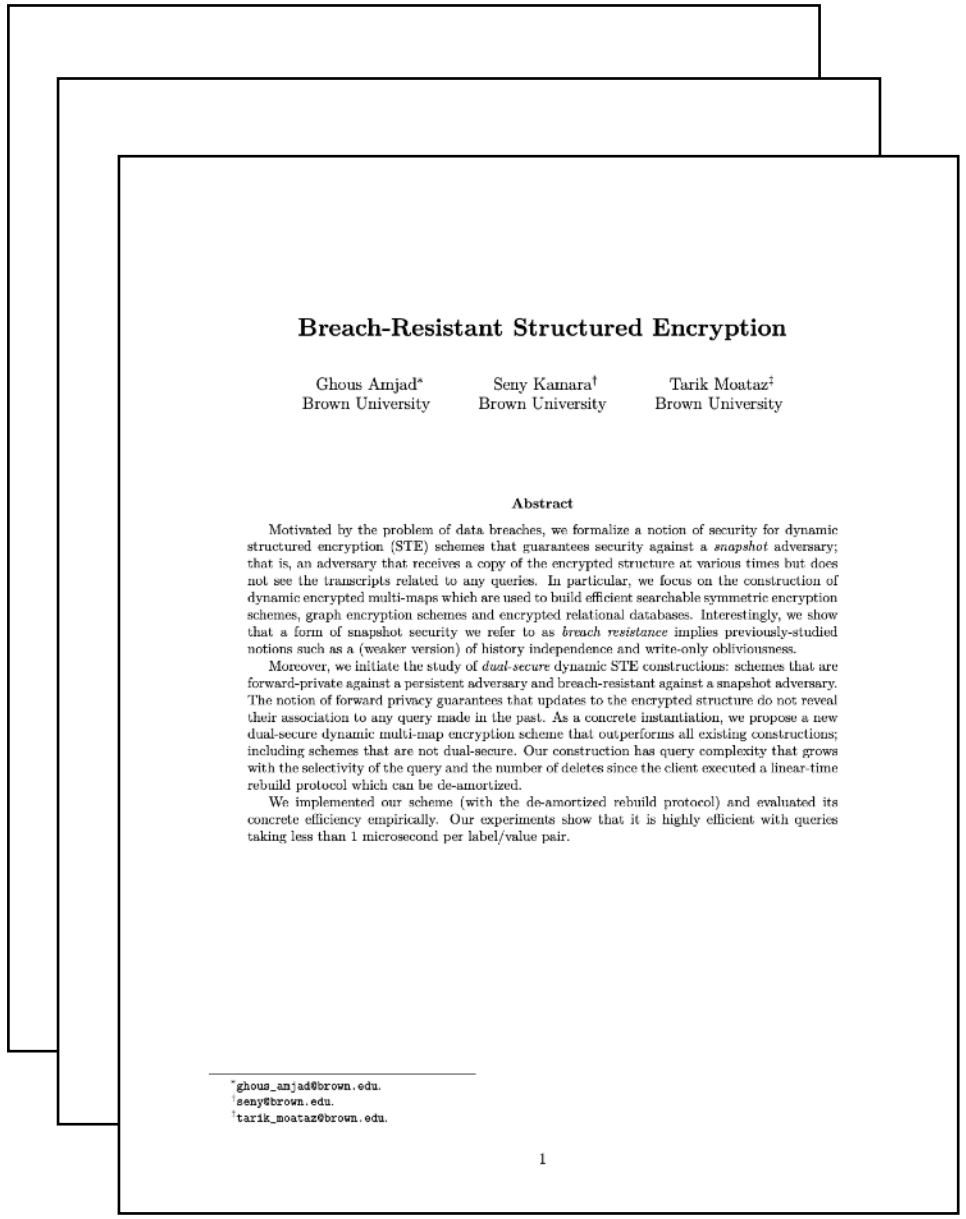
The Range Cryptanalysis Papers

- [Kellaris-Kollios-Nissim-O'Neill'16]
  - exploits response identity or volume
  - assumes uniform queries
- [Lacharite-Minaud-Patterson'18,]
  - exploits response identity & assumes density
- [Grubbs-LMP'18]
  - exploits volume (response length)
  - assumes (variants of) all response lengths occur
- [Kornaropoulos-Papamanthou-Tamassia'19]
  - exploits volume (response length) & query equality
  - assumes unique values or additional leakage

C: I told you! Look at all these attacks!

A: Attacks are interesting but many are not practical

Q: What happens in the snapshot model?



Breach-Resistant Structured Encryption

Ghous Amjad* Brown University
Seny Kamara† Brown University
Tarik Moataz‡ Brown University

Abstract

Motivated by the problem of data breaches, we formalize a notion of security for dynamic structured encryption (STE) schemes that guarantees security against a *snapshot* adversary; that is, an adversary that receives a copy of the encrypted structure at various times but does not see the transcripts related to any queries. In particular, we focus on the construction of dynamic encrypted multi-maps which are used to build efficient searchable symmetric encryption schemes, graph encryption schemes and encrypted relational databases. Interestingly, we show that a form of snapshot security we refer to as *breach resistance* implies previously-studied notions such as a (weaker version) of history independence and write-only obliviousness.

Moreover, we initiate the study of *dual-secure* dynamic STE constructions: schemes that are forward-private against a persistent adversary and breach-resistant against a snapshot adversary. The notion of forward privacy guarantees that updates to the encrypted structure do not reveal their association to any query made in the past. As a concrete instantiation, we propose a new dual-secure dynamic multi-map encryption scheme that outperforms all existing constructions; including schemes that are not dual-secure. Our construction has query complexity that grows with the selectivity of the query and the number of deletes since the client executed a linear-time rebuild protocol which can be de-amortized.

We implemented our scheme (with the de-amortized rebuild protocol) and evaluated its concrete efficiency empirically. Our experiments show that it is highly efficient with queries taking less than 1 microsecond per label/value pair.

*ghous_amjad@brown.edu.
†seny@brown.edu.
‡tarik_moataz@brown.edu.

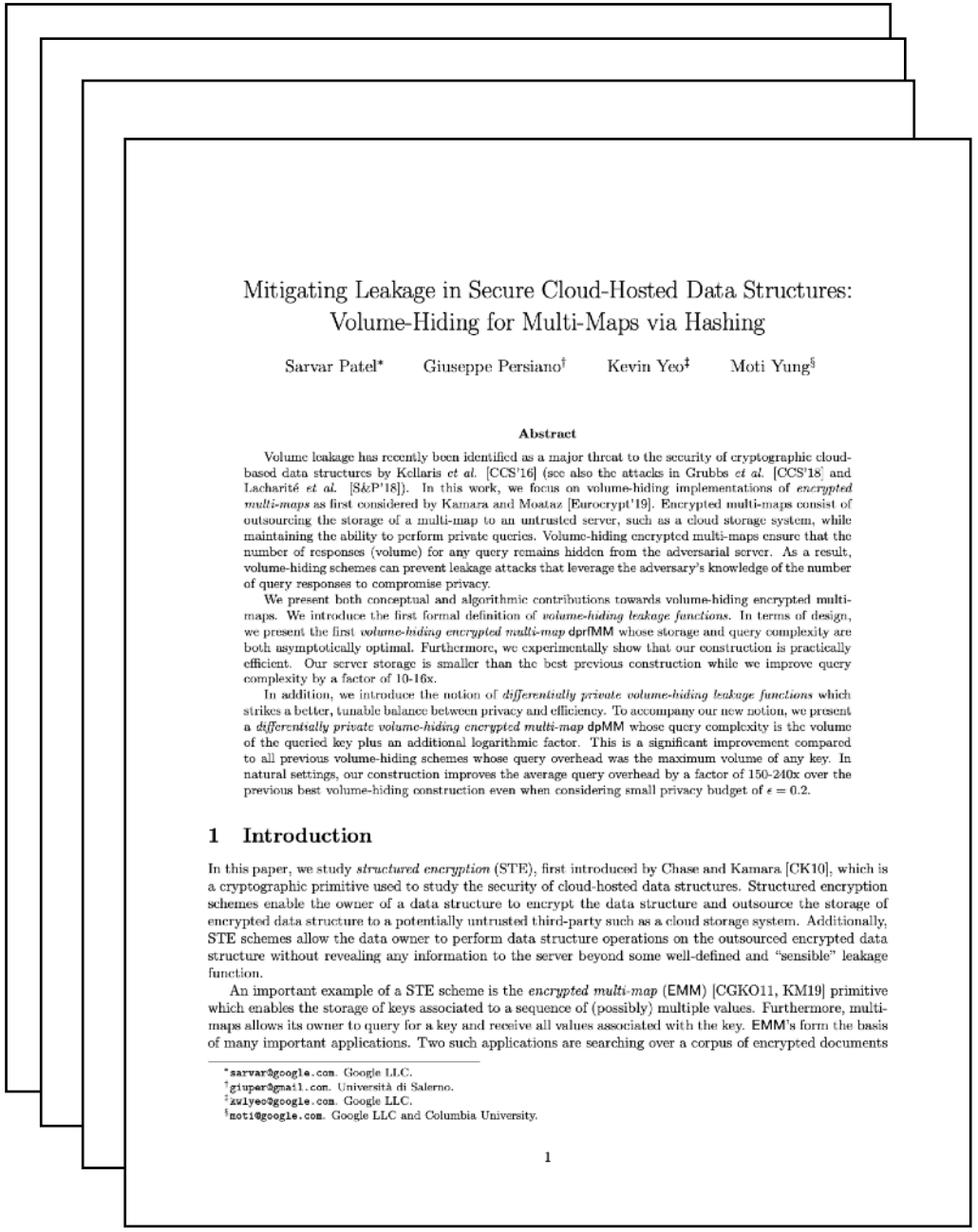
1

The Snapshot Papers

- [Lewi-Wu'16]
  - formalized ORE in snapshot model
  - … schemes in snapshot model
- [Grubbs-Ristenpart-Shmatikov'17]
  - is snapshot model realistic for databases?
- [Amjad-K.-Moataz'19]
  - formalized STE in snapshot model
  - *zero-leakage* dynamic schemes in snapshot model

C: Well…there's still leakage in the persistent model!

Q: Can we suppress leakage in persistent model?



Mitigating Leakage in Secure Cloud-Hosted Data Structures: Volume-Hiding for Multi-Maps via Hashing

Sarvar Patel* Giuseppe Persiano† Kevin Yeo‡ Moti Yung§

Abstract

Volume leakage has recently been identified as a major threat to the security of cryptographic cloud-based data structures by Kellaris *et al.* [CCS'16] (see also the attacks in Grubbs *et al.* [CCS'18] and Lacharité *et al.* [S&P'18]). In this work, we focus on volume-hiding implementations of *encrypted multi-maps* as first considered by Kamara and Moataz [Eurocrypt'19]. Encrypted multi-maps consist of outsourcing the storage of a multi-map to an untrusted server, such as a cloud storage system, while maintaining the ability to perform private queries. Volume-hiding encrypted multi-maps ensure that the number of responses (volume) for any query remains hidden from the adversarial server. As a result, volume-hiding schemes can prevent leakage attacks that leverage the adversary's knowledge of the number of query responses to compromise privacy.

We present both conceptual and algorithmic contributions towards volume-hiding encrypted multi-maps. We introduce the first formal definition of *volume-hiding leakage functions*. In terms of design, we present the first *volume-hiding encrypted multi-map* dprfMM whose storage and query complexity are both asymptotically optimal. Furthermore, we experimentally show that our construction is practically efficient. Our server storage is smaller than the best previous construction while we improve query complexity by a factor of 10-16x.

In addition, we introduce the notion of *differentially private volume-hiding leakage functions* which strikes a better, tunable balance between privacy and efficiency. To accompany our new notion, we present a *differentially private volume-hiding encrypted multi-map* dpMM whose query complexity is the volume of the queried key plus an additional logarithmic factor. This is a significant improvement compared to all previous volume-hiding schemes whose query overhead was the maximum volume of any key. In natural settings, our construction improves the average query overhead by a factor of 150-240x over the previous best volume-hiding construction even when considering small privacy budget of $\epsilon = 0.2$.

1 Introduction

In this paper, we study *structured encryption* (STE), first introduced by Chase and Kamara [CK10], which is a cryptographic primitive used to study the security of cloud-hosted data structures. Structured encryption schemes enable the owner of a data structure to encrypt the data structure and outsource the storage of encrypted data structure to a potentially untrusted third-party such as a cloud storage system. Additionally, STE schemes allow the data owner to perform data structure operations on the outsourced encrypted data structure without revealing any information to the server beyond some well-defined and "sensible" leakage function.

An important example of a STE scheme is the *encrypted multi-map* (EMM) [CGKO11, KM19] primitive which enables the storage of keys associated to a sequence of (possibly) multiple values. Furthermore, multi-maps allows its owner to query for a key and receive all values associated with the key. EMM's form the basis of many important applications. Two such applications are searching over a corpus of encrypted documents

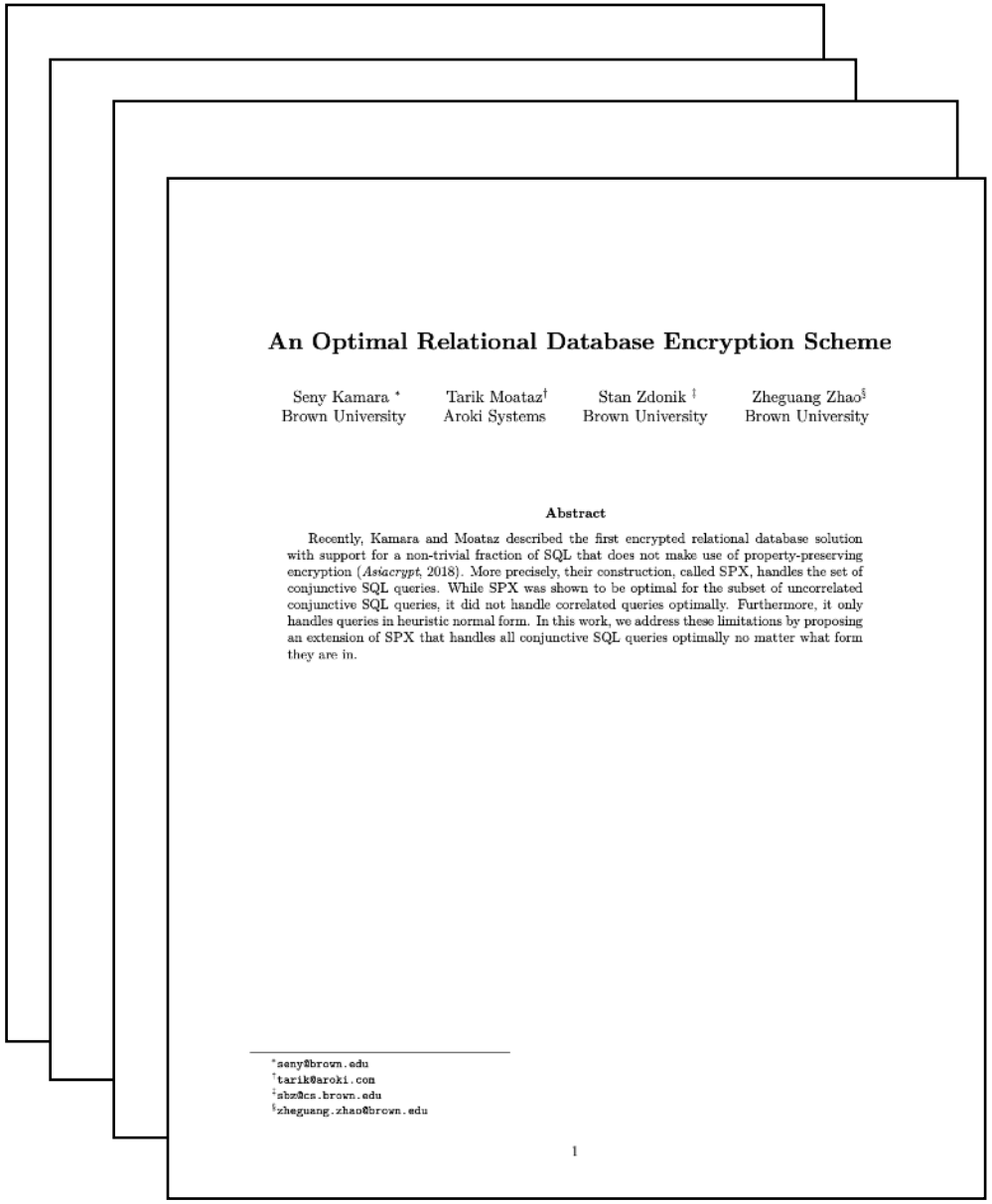
*sarvar@google.com. Google LLC.
†giuper@gmail.com. Università di Salerno.
‡kwlyeo@google.com. Google LLC.
§moti@google.com. Google LLC and Columbia University.

The Leakage Suppression Papers

- [K.-Moataz-Ohrimenko'18]
  - suppresses query eq. for static encrypted structures
- [K.-Moataz'19]
  - suppresses volume for static encrypted structures
- [George-K.-Moataz'21]
  - suppresses query eq. for dynamic encrypted structures
- [Patel-Persiano-Yeo-Yung'19]
  - volume-hiding EMM with optimal storage & queries

C: Who cares if you can search? Can you do databases?

Q: Can we encrypted databases with O(OPT) queries?

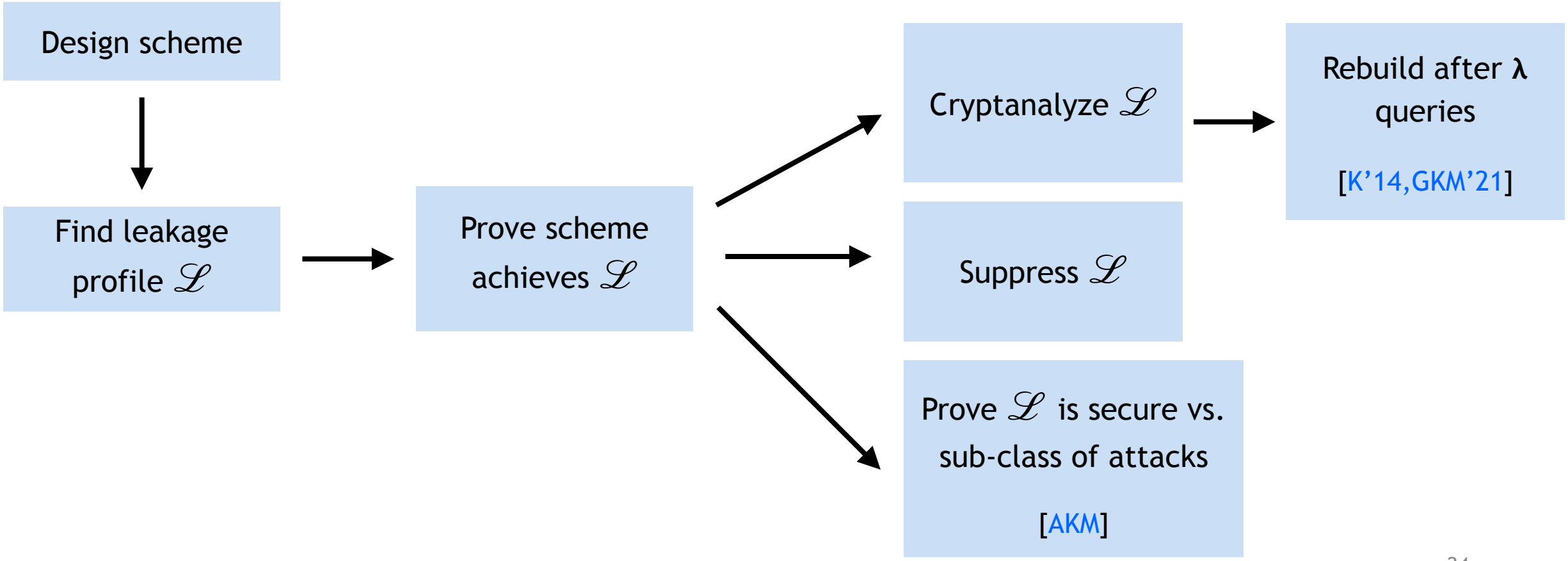


The Database Papers

- [Popa-Redfield-Zeldovich-Balakrishnan'11]
  - SQL using det. and order-preserving encryption
  - Practical attacks with high recovery rates
- [Chase-K.'10]
  - introduced & formalized structured encryption
- [K.-Moataz'18, K.-Moataz-Zdonik-Zhang'20]
  - Large subset of SQL using structured encryption
  - Complex leakage
- [K.-Moataz]
  - MQL using structured encryption

Q: So what is the result of all this work?

A Unique Design & Analysis Framework



Real Technology!

MongoDB. Products Solutions Resources Company Pricing Sign In Try Free

Rich querying capabilities on encrypted data

MongoDB is the only database provider that enables processing of expressive queries on randomly encrypted data. Data is never in cleartext in the database, but MongoDB can still process queries and execute operations on the server side.

What Am I Trying to Say?

- The field has come a really long way…
  - …in the face of criticism & pushback

- Went from interesting research problem to a real technology

- This was achieved thanks to the Dreamers!
  - The ones who saw the long term vision
  - The ones who pursued science despite the critics
  - The ones who stuck to it

What Am I Trying to Say?

- There are a lot more problems to solve
  - …and more criticism & pushback to come

- But we should keep pushing forward…
  - …and never get discouraged!

Thank you!