Encrypted Search

Seny Kamara
Microsoft Research

Abstract

A large amount of sensitive and private information is being collected and stored in corporate and government systems throughout the world. The importance and utility of this data, however, prevents it from being encrypted since we would lose the ability to search over it. The area of encrypted search, which is concerned with the design and analysis of cryptographic techniques for searching on encrypted data, could solve this dilemma and fundamentally transform the way we store and process information.

1 Introduction

As we produce increasingly larger amounts of data, several important trends have started to emerge. The benefits of Big Data—including advances in machine learning, social sciences, marketing and analytics—are well-publicized, but the many problems it presents have received less attention from the public at large. As data gets bigger, it becomes more intrusive and privacy-sensitive. And as governments and corporations choose to produce and store more data about their citizens and customers—even if only for national security purposes or to improve their services—the amount of private information available about each and every one of us keeps growing. Big datasets include electronic medical records (EMR), location data, browsing histories, search queries, pictures, phone call metadata, demographic information, emails, social networks and, more and more, voicemails and television streaming/viewing habits. While some datasets are clearly more sensitive than others, even seemingly innocuous ones can be used or combined to infer private information.

This article surveys recent advances in cryptography that address an inherent conflict between two important trends that occur when data gets bigger: *on one hand our increased reliance on search and on the other our growing inability to properly secure data.*

Searching Big Data. As data gets bigger, it becomes harder to work with and we become more reliant on fast searching algorithms and data structures. Search is arguably the most fundamental operation in Computer Science and it is the main objective of core areas such as Databases (in the context of structured data) and Information Retrieval (in the context of unstructured data). In the mid-90s, unstructured data started growing in size and this motivated the development of search engines for the Web and desktop computers. By the end of that decade, search engines had become ubiquitous and integral part of every system. Today, one cannot imagine deploying any kind of application without a search box.

Securing Big Data. But as data gets bigger, it also becomes harder to secure. Consider that the lowest estimate for the storage capacity of the NSA Bluffdale datacenter is 2 exobytes (i.e., 2000 petabytes) or that Facebook has approximately 300 petabytes of videos and pictures. Imagine the challenges in securing datasets of such sensitivity and size against nation states, organized crime, hackers and rogue employees. But there are serious security challenges even on the lower-end of the Big Data spectrum. Consider electronic medical records, which include large images like x-rays, CAT scans and MRIs. A medium-to-large hospital can generate around 20 terabytes in EMRs per year and has to remain compliant with privacy regulations all the while allowing the records to be shared among different users and institutions and across multiple devices.

As many smaller businesses and institutions start to grapple with Big Data, many are thinking about outsourcing their storage requirements to Cloud providers who have more expertise and experience. But Cloud storage introduces a whole new set of security and privacy challenges. If the outsourced data is sensitive, regulated or mission-critical, then its disclosure to a cloud provider may be illegal or potentially damaging to the data owner.

Encrypted search. So on one hand, as data gets bigger it becomes more intrusive and it is even more critical to secure. On the other, it becomes more difficult to protect. This is exacerbated by the fact that the traditional mechanisms for securing data, which consist of encryption for confidentiality and digital signatures for integrity, are not applicable to Big Data since they naturally eliminate the ability to operate on it and, in particular, to search through it. In other words, our reliance on search, which is indispensable when working with Big Data, obviates the most important tools at our disposal to secure data.

To address this fundamental conflict between Big Data security and search, the area of encrypted search has recently emerged as one of the most exciting and potentially impactful topics in cryptography. Encrypted search is concerned with the design of encryption schemes that support various forms of search on encrypted data. Several approaches have been developed to address this problem, including solutions based on fully-homomorphic encryption (FHE), oblivious RAMs (ORAM), property-preserving encryption, functional encryption and structured encryption.

2 A Variety of Approaches

Encrypted search solutions blend non-trivial ideas from cryptography, data structures, algorithms, information retrieval and databases. Currently, the state-of-the-art constructions achieve different tradeoffs between security, efficiency and query expressiveness.

Security & expressiveness vs. efficiency. When maximizing security and query expressiveness at the expense of efficiency, the best solutions are based on ORAMs and FHE. The best solutions in this regime can support arbitrary queries on encrypted data without leaking any information. Unfortunately, these approaches are too inefficient to be of practical interest. FHE-based solutions are impractical not only due to the present cost of executing homomorphic operations but also due to the fact that current FHE-based encrypted search requires linear time, which is not feasible for Big Data. And while ORAM-based solutions perform a lot better, they are still far from practical for large datasets.

Efficiency & expressiveness vs. security. When maximizing efficiency and query expressiveness at the expense of security, the best solutions support large classes of SQL queries on encrypted relational databases. The solutions in this regime are currently based on property-preserving encryption (e.g., deterministic encryption and order-preserving encryption) and, as such, are very efficient but leak a considerable amount of information (e.g., simple frequency analysis can often be used to attack such systems). State-of-the-art implementations of these solutions are competitive with commercial database systems.

Security & efficiency vs. expressiveness. When maximizing security and efficiency at the expense of query expressiveness, the best solutions support Boolean keyword search on encrypted text as well as various queries on encrypted graphs. The solutions in this regime are based on structured encryption and, therefore, are very efficient and leak only a small amount of information. Keyword search can be achieved in sub-linear and even optimal time (i.e., search time that is a linear function of the number of documents that contain the keyword) and the state-of-the-art implementations are only a few times more expensive than commercial systems. Currently the main limitation of these solutions is a lack of query expressiveness which makes them more suitable for NoSQL databases and applications that work with unstructured data like Web mail, desktop search engines and cloud storage services like Dropbox and OneDrive.

3 Structured Encryption

Despite its current limitations in terms of query expressiveness, the most promising approach to encrypted search is structured encryption as it provides the right balance between efficiency and security. Structured encryption schemes encrypt data structures in such a way that they can be privately queried. Typically, a specific scheme supports a particular kind of data structure so, for example, a set encryption scheme encrypts sets with support for membership queries. Similarly, a graph encryption scheme encrypts graphs with support for various graph queries like neighbor queries or shortest distance queries. Given the ubiquity and importance of data structures in Computer Science, it should come as no surprise that structured encryption would have a large number of potential applications.

Encrypted search via structured encryption. The original motivation for structured encryption was the problem of encrypted search. Consider the case of a client that wants to securely store a dataset (e.g., an email inbox, a Dropbox folder, a collection of EMRs) on an untrusted server. One can use structured encryption to design an encrypted search solution as follows. Structured encryption schemes that are specialized for keyword search are usually called searchable symmetric encryption schemes. The client first builds a data structure that supports fast search queries over the dataset; that is, with search time that is sub-linear in the size of the data. In the case of keyword search, for example, this could be an inverted index. It then encrypts the search structure with an appropriate structured encryption scheme. In our example, this would be an index encryption scheme that supports sub-linear keyword search. The dataset itself can then be encrypted using any standard encryption scheme (e.g., the Advanced Encryption Standard). Note that each document/record in the dataset is encrypted individually. Since the data and search structure are both encrypted, the underlying dataset is protected and can be safely stored on the server. To search the encrypted data, the client just needs to query the encrypted structure. Depending on

the particular construction, these queries can take the form of an interactive two-party protocol or of a simple non-interactive step. In either case, the client receives the result of its query and can then retrieve the appropriate encrypted documents.

4 Defining and Analyzing Security

The methodology that is used in modern cryptography to analyze and assess the security of cryptographic algorithms and protocols is called the reductionist security paradigm (sometimes also called provable security). In this framework, one first proposes a formal security definition and then describes a reduction from the cryptosystems security to an underlying assumption (e.g., that factoring large integers is hard). The cryptosystem achieves the security guarantees captured by the security definition if the underlying assumption is true.

Formalizing security. Early efforts to formalize the security of encrypted search and structured encryption did not capture all the subtleties of how an adversary can interact with an encrypted search solution. One issue was that unlike standard encryption schemes, where the adversary holds a target ciphertext and can receive encryptions of chosen messages, a structured encryption schemes adversary is provided a target ciphertext and can see the execution of a search on a chosen query. Moreover, the ability to witness the search can be much more helpful to the adversary than the ability to just see other ciphertexts. This is particularly the case if the search operations leak information about the data and/or the query. As a result, it was realized that the security definitions for standard encryption schemes were not strong enough for searchable encryption.

However, more subtle issues remained. It was eventually understood that a proper security definition for structured encryption should also capture the adversarys ability to choose queries as a function of previous queries, results and of the leakage that it received. Time was also required to fully appreciate and formalize the leakage that these new types of encryption schemes presented.

Leaky cryptography. Interestingly, leaky cryptographic primitives have hardly received any attention from the cryptography community up to this point. This makes sense since the goal of encryption is ostensibly to protect data and to eliminate all possible leakage. What makes structured encryption different is its explicit goal of trading off leakage for efficiency. The willingness to make such a tradeoff and the potential impact that practical and scalable solutions could have, provide a natural motivation to consider and explore leakage in more depth.

These advances in our understanding of the security requirements of structured encryption eventually led to the notion of leaky adaptive security which, roughly speaking, guarantees that a structured encryption scheme leaks nothing beyond what is explicitly allowed by a given leakage profileeven to an adversary that can choose its queries as a function of previous queries, results and leakage. This notion of security is interesting for several reasons. First, it seems to capture our intuition about what a secure encrypted search solution should provide while being flexible enough to allow for leakage versus efficiency tradeoffs. Second, it is general enough to capture the security of many other important cryptographic primitives like ORAM, private information retrieval and garbled gates.

5 Attacking Structured Encryption

Leaky adaptive security provides a way to analyze the security of a structured encryption scheme with respect to a given leakage profile. Another way to put it is that it gives us a way to bound the leakage of a solution. What it does not provide is a way to understand and predict the impact of that leakage. So, while we can show that a scheme leaks nothing beyond a given leakage profile, we currently have no theoretical framework to analyze and understand this leakage.

In light of this, a natural and alternative approach is to improve our empirical understanding of leakage by studying the attacks that make use of it. These attacks, referred to as inference attacks, combine leakage and auxiliary information (e.g., publicly available information like census data or language statistics) to recover information about the data and/or queries. Inference attacks can be viewed as more sophisticated variants of the traditional frequency analysis which has been known since the 9th century and was used to break classic ciphers.

The study of inference attacks, however, has additional benefits beyond improving our understanding of leakage. In fact, inference attacks can be used to set the parameters of structured encryption schemes similarly to how various number-theoretic algorithms (e.g., the number field sieve) are used to set the security parameters of traditional encryption schemes (e.g., RSA). This is because to mitigate the effect of certain leakage profiles, one can always re-encrypt the dataset and the structure after a certain number of queries. This number is called the query capacity and the fundamental question here is how large should it be, i.e., how many queries should we perform before re-encryption? On one hand, since re-encryption is expensive we want to maximize the query capacity. On the other, since queries leak information we want to minimize it. Inference attacks can help us find the right balance between these constraints. By setting the query capacity just low enough to prevent the best-known inference attacks, we can maximize the query capacity while still guaranteeing an empirically-meaningful level of security.

6 Challenges Ahead

There are many interesting and important challenges left to address in structured encryption and, more generally, in encrypted search. It is imperative to improve our understanding of leakage. This can come either through new theoretical models that capture and analyze leakage or from new insights gained through the empirical study of inference attacks. Also, establishing lower bounds on the search and storage complexities of structured encryption would be of great value. Intuitively, it seems that ideas from the field of data structure lower bounds might be helpful here.

On the constructive end, designing schemes that achieve better tradeoffs between efficiency, leakage and expressiveness is extremely important. Of particular interest is the design of structured encryption schemes that support SQL queries on encrypted relational databases with less leakage than the current proposals based on property-preserving encryption. New and improved constructions for semi-structured data like XML data would also prove useful. Other important problems in this direction include developing schemes with better support for range queries and ranked searches. The study of non-text data is of particular importance. For example, the design of graph encryption schemes with support for various graph queries could prove to be very impactful as graph databases start to gain more and more prominence in the context of Big Data.

There are also very exciting problems on the engineering end of the spectrum. For example, implementing and engineering structured encryption to work with large-scale systems and services

like Web mail, EMRs and commercial databases would be very impactful and provide invaluable insights into how to best optimize the cryptographic schemes. For example, much of the work that focused on engineering encrypted search systems based on structured encryption uncovered new bottlenecks and opportunities for optimization that theory alone did not find. This recently motivated the design of a new generation of schemes that make better use of the memory hierarchy and that are parallelizable.

7 Further Reading

Here, we mention only the works discussed in this article and recommend the survey of Bösch, Hartel, Jonker and Peter [3] for a comprehensive overview of the research literature.

The first work to explicitly consider the problem of searching on encrypted data is a paper by Song, Wagner and Perrig from 2001 [13]. The notion of leaky adaptive security and the first practical encrypted search solutions were proposed in 2006 in a paper by Curtmola, Garay, Kamara and Ostrovsky [7]. Structured encryption and graph encryption were introduced in a 2010 paper by Chase and Kamara [6]. Cash, Jarecki, Jutla, Krawczyk, Rosu and Steiner showed how to extend the approach of [7] to handle boolean queries efficiently [5]. Kamara and Papamanthou in [11] and Cash, Jaeger, Jutla, Krawczyk, Rosu and Steiner in [4] proposed SSE constructions that are parallel and I/O-efficient. The first inference attack was described in 2012 by Islam, Kuzu and Kantarcioglu [10].

ORAMs were introduced by Goldreich and Ostrovsky in [9]. The first FHE scheme was proposed by Gentry in 2009 [8]. Deterministic and order-preserving encryption were first studied formally by Bellare, Boldyreva and O'Neill in [1] and Boldyreva, Chenette, Lee and O' Neill in [2]. The CryptDB system of Popa, Redfield, Zeldovich and Balakrishnan [12] makes use of deterministic and order-preserving encryption to support SQL queries over encrypted relational databases.

References

[1] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *Advances in Cryptology – CRYPTO '07*, Lecture Notes in Computer Science, pages 535–552. Springer, 2007.

[2] A. Boldyreva, N. Chenette, Y. Lee, and A. O'neill. Order-preserving symmetric encryption. In *Advances in Cryptology - EUROCRYPT 2009*, pages 224–241, 2009.

[3] C. Bösch, P. Hartel, W. Jonker, and A. Peter. A survey of provably secure searchable encryption. *ACM Computing Surveys (CSUR)*, 47(2):18, 2014.

[4] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner. Dynamic searchable encryption in very-large databases: Data structures and implementation. In *Network and Distributed System Security Symposium (NDSS '14)*, 2014.

[5] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner. Highly-scalable searchable symmetric encryption with support for boolean queries. In *Advances in Cryptology - CRYPTO '13*. Springer, 2013.

[6] M. Chase and S. Kamara. Structured encryption and controlled disclosure. In *Advances in Cryptology - ASIACRYPT '10*, volume 6477 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2010.

[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *ACM Conference on Computer and Communications Security (CCS '06)*, pages 79–88. ACM, 2006.

[8] C. Gentry. Fully homomorphic encryption using ideal lattices. In *ACM Symposium on Theory of Computing (STOC '09)*, pages 169–178. ACM Press, 2009.

[9] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious RAMs. *Journal of the ACM*, 43(3):431–473, 1996.

[10] M. Saiful Islam, M. Kuzu, and M. Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. In *Network and Distributed System Security Symposium (NDSS '12)*, 2012.

[11] S. Kamara and C. Papamanthou. Parallel and dynamic searchable symmetric encryption. In *Financial Cryptography and Data Security (FC '13)*, 2013.

[12] R. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 85–100. ACM, 2011.

[13] D. Song, D. Wagner, and A. Perrig. Practical techniques for searching on encrypted data. In *IEEE Symposium on Research in Security and Privacy*, pages 44–55. IEEE Computer Society, 2000.