RESEARCH ARTICLE OPEN ACCESS

Significant Approaches to Computation on Encrypted Data

Irene Getzi S
Department of MCA
Jyoti Nivas College
Bangalore - India

ABSTRACT
Cloud computing bring about a major shift in computing. The control over data and operations from the client organization is now shifted to their cloud providers. The basic tasks like storage, retrieval and maintenance of data become the responsibility of the cloud service provider and not the user. As more and more security breaches happening all over the world in different forms it is necessary to protect the data from the adversaries. The most difficult adversary to protect against in a public cloud is the cloud provider itself. It controls everything: the data center, the network, the operating system the hardware and the users. Thus it becomes rampant to store the data in the encrypted format that provides confidentiality and privacy. However as soon as data is encrypted it becomes seemingly impossible to perform computations on the data without decrypting it. The question of computing on encrypted data (COED) has instigates a number of fascinating techniques as well as deep open problems. It remains a driving force behind much of the most exciting research in cryptography today. This paper presents some of the underlying technologies and overview of the work done in this regard.
Keywords :- Computations on encrypted data, COED, functional encryption, symmetric searchable encryption, secure searches, homomorphic encryption

I. INTRODUCTION

We live in the era of information explosion and have witnessed the trend of leveraging cloud-based services for large scale content storage, processing, and distribution. "Data-as-a-Service" (DaaS) applications are on arise and various business systems are trying to evolve and benefit from cloud services. Though the benefits of cloud computing is clear, the cloud based systems poses an added level of risk because the essential services are now outsourced to a third party. Thus it is necessary to develop a proper security mechanism for cloud implementations to reap the benefits it offers. As the cloud server itself cannot be trusted, we need a secured technology that protects our data and applications without relying on or revealing information to the untrusted server.

One possible approach to enforce protection to the data without relying on cloud servers could be to encrypt data before outsourcing to the untrusted server and disclose the corresponding decryption keys only to authorized users. This reduces security and privacy risks by hiding all information about the plaintext data. Encryption makes it impossible for both insiders and outsiders to access the data without the keys but at the same time makes traditional data utilization services such as searching, retrieving and performing computations on the data a difficult task.

Data sets are becoming very large nowadays that it becomes difficult to keep a local copy of data in the client location to perform analysis. Often, it is also the case that the data cannot be aggregated at a single repository due to mutual privacy concerns.

A problem with the traditional data retrieval or computational system is that, when a query is given on the encrypted data, it lets the server decrypt the data, runs the query on the server side, and sends only the results back to the user. This allows the server to learn the data being queried or computed and hence makes encryption less useful. The security mechanism adapted should not leak any information on the access or the search to the server. In short, the system should allow the user to perform complicated processing of data without being able to see it.

Computation on Encrypted Data (COED) seeks to develop methods that allow computing with encrypted data without first decrypting it, making it more difficult for adversaries to get to know the information. These technologies are becoming increasingly well developed, and some have found commercial application. In this paper, we outline some of these technologies that support COED; these are technologies which support Encrypted Search such as Searchable Symmetric Encryption

II. BACKGROUND STUDY

COED complements the cryptographic protocols that protect data in transit and at storage by offering protection during computation. COED technologies offer end-to-end protection to sensitive information by allowing data to stay encrypted at all times even when performing computation.

The most basic form of computation on encrypted data is to perform a search on it as *Search* is the primary way to access the stored data. The problem of encrypted search is now of

interest to many sub-fields in computer science as well as for many privacy preserving applications related to business, health-care and to governments.

The problem of searching on encrypted data was first considered explicitly by Song, Wagner and Perrig in [1]. Various searchable encryption schemes were proposed that uses technologies like property-preserving encryption [6], Functional encryption [7], symmetric searchable encryption [2, 4], and homomorphic encryption [9, 10].

Secured searches are a well-studied concept and it offers a very rich set of search queries on encrypted data such as boolean queries, multi-key word search queries, and range queries. Ongoing researches are happening on searching through very large scale datasets and multicloud environments.

In order to reap the full benefits offered by cloud, secured searching alone may not be sufficient. Data mining or analytics application may require basic to complex computations to be performed on the encrypted data. Homomorphic encryption, secure multiparty computation and Oblivious RAM can offer significant results in this regard.

III. SIGNIFICANT COED TECHNOLOGIES WITH CONTROLLED LEAKEGE

There are a number of ways to compute on encrypted data and each one is suitable for a specialized setting. The different tradeoffs considered are functionality, security, and efficiency. While homomorphic encryption and Oblivious RAM can offer better security without leakage it cannot provide a practical solution. When fully homomorphic or ORAM based schemes cannot provide a practical solution, we can take advantage of controlled leakage constructions. This section provides an overview on some of the competing technologies used for computation on encrypted data with controlled leakage.

A. Property-Preserving Encryption

Property-preserving encryption (PPE) is a specialized encryption scheme that can be used in secure outsourced storage that intentionally leaks certain properties of the underlying message. PPE-based solutions can achieve fast sub-linear time server-side encrypted search. But they do have some limitations such as it leaks quite a bit of information to the server about the data collection [6].

There are different types of PPE schemes that each leak different properties. The two main examples are deterministic encryption (DE), which preserves the equality property, and order preserving encryption (OPE), which preserves order comparison.

Though deterministic encryption scheme offers a faster solution, it has the limitation that it always encrypts the same message to the same ciphertext. Also the server can learn if the client is repeating a search a not. Another issue occurs when the scheme uses public-key; the data is encrypted by client's public key that is available to the server. The server can then mount a dictionary attack on the encrypted database.

Order-preserving encryption is, apparently, a method of encrypting data so that it's possible to make efficient inequality comparisons on the encrypted items without decrypting them. Such a solution can be used when the data has high min-entropy, which is a way of saying that the data looks random to the server.

B. Functional Encryption

For many cloud services applications, there is often a need to specify a decryption policy in the ciphertext and only individuals who satisfy the policy can be allowed to decrypt. i.e. we may want to only give access to a function/part of the plaintext, depending on the authorization rules of a user. A functional encryption system, with a decryption key allows a user to learn a function of the encrypted data. i.e. in a functional encryption system for functionality F, an authority holding a master secret key can generate a key that enables the computation of the function on encrypted data.

The notion of FE was first described by Sahai and Waters in a talk [17] and Later Boneh et al.[7] formalized its construction is based Identity Based Encryption (IBE) where data gets encrypted using receiver's identification such as passwords and the receiver can search on the ciphertext using secret key generated with his ID.

Various other properties were considered and give rise what is called Attribute-Based Encryption (ABE). ABE can be regarded as a generalization of IBE. It decomposes the user identity as a set of attributes, and enables every user to have its unique attributes set. The encryptor can pick an attributes set and generate an access control policy associated this attributes set, then he can use encrypt data with this policy. A decryptor whose attributes set satisfy the policy have the permission to decrypt the ciphertext. Its flexible access control policy and multi-user support property turned out to be extremely suitable for cloud architecture.

The schemes based on functional encryption offered better security and access control mechanism but are slower than the PPE-based approach. The encrypted database by itself does not reveal much useful information to the server since keywords are encrypted using a *randomized* (identity-based) encryption scheme. But the problem with this approach is that as with any public-key encrypted search solution, the server has both the ability to create the index file with the user's public key and to search over them.

Hence symmetric key setting was introduced that may be well suitable for this setting because only client can generate encrypted index and search tokens.

C. Symmetric Searchable Encryption

The Symmetric searchable encryption scheme (SSE) employs a prebuilt search index encrypted by the client's secret key that is sent to the outsourced server along with the encrypted documents. The scheme lets users with appropriate tokens securely search over the encrypted data through keywords without decrypting it.

The symmetric searchable scheme proposed by Song in [1] uses Steam cipher method for encryption and supports sequential search method that may allow attackers to get the information of keywords using a statistical analysis on the searching result.

The works that followed in this area proposed index-based search schemes and focused on how to build an efficient and secure index for the data files. Different index structures are proposed to extend the functionality of SSE that supports search queries on matrix data, labeled data, neighbor and adjacency queries on graph data multi-keyword boolean search queries.

Subsequently, the inverted index approaches were proposed in [2, 4] that reduced the search time to sub-linear time. While the inverted index approach yields the efficient search time, it is not well-suited to handle dynamic collections. The above-mentioned schemes mainly used only static indexes that make it difficult for the user to dynamically add or update to the existing index or document collection. That requires the index file to be rebuilt.

The notion of dynamic updates was introduced by Kamara et al. in the [2], which allows dynamic addition and deletion of index entries. The *Dynamic SSE* schemes require additional data structures such as search tables and arrays and thus making their construction very complex and difficult to implement.

The major limitation with the SSE schemes is that it leaks certain information to the underlying server such as the access and search patterns. In fact, by observing enough search results the server could use some sophisticated statistical attack to infer something about the client's queries and data. This limitation can be overcome with the Oblivious RAM or homomorphic technologies.

IV. SIGNIFICANT COED TECHNOLOGIES WITHOUT LEAKEGE

A. Oblivious RAM (ORAM)

ORAM-based solution can offer optimal level of security ORAM-based solution can offer optimal level of security guarantee without any leakage. Originally proposed by Goldreich and Ostrovsky [3], ORAM is a cryptographic construction that allows a client to read and write to memory without the memory device knowing which locations are being accessed. Particularly, the sequence of physical addresses accessed is independent of the actual data that the user is accessing. To achieve this, existing ORAM constructions continuously re-encrypt and reshuffle data blocks on the storage server.

Thus a client can access the encrypted data residing on an untrusted storage server, while completely hiding the access patterns to storage. Unfortunately, this approach requires many rounds of interaction and has a high overhead that makes ORAM inefficient and unacceptable especially for large scale data outsourcing applications.

B. Homomorphic Encryption

Homomorphic encryption (HE) is a cryptographic construction that allows computations to be carried out on encrypted data, such that the generated encrypted result, when decrypted, matches the result of operations performed on the clear text. It allows mathematical operations to be performed on encrypted without compromising the encryption. In 2009, Craig Gentry introduced the first Fully Homomorphic Encryption (FHE) scheme that was based on lattice-based cryptosystem. However, Gentry's scheme proved unfeasible in practice due to a massive overhead in both computation and memory cost.

A quite lot of work is being carried out to increase the efficiency of the scheme. The homomorphic properties mainly focus on additions and multiplications because, *in theory*, any operation can be brought down to a binary circuit at its lower level.

The existing public-key systems, such as ElGamal [12] and Pallier [6], support only one homomorphic operation. A doubly-homomorphic system proposed in [15] can be applied only to compute few boolean operations as the ciphertext size doubled at every step. As a result, one could only perform few boolean operations before the ciphertext size became unmanageable. Recently constructions based on elliptic curves are gaining momentum and seems to produce better results.

C. Secure Multiparty Computation

Computation on encrypted data using HE schemes slows down by nearly 10 orders of magnitude, making it infeasible. A related research area is secure multiparty computation (SMC), in which multiple entities can jointly perform computations while maintaining the privacy of each entity's data. SMC protocols incur significant overhead, but comparatively less than HE schemes.

Secure Multiparty Computation (SMC) allows multiple parties to perform computation on their private data to evaluate some function, without revealing their private data. Only the result of the computation will be available to all the parties. This privacy-preserving technique can be used when multiple cloud providers or multiple cloud users jointly compute some function of their private data inputs.

Fully homomorphic encryption (FHE) enables secure computation over the encrypted data of a single party. This can be extended to multiple parties using multiparty computation. Recently various schemes [10, 13, 14] are proposed that combine the functionalities of homomorphic encryption and multiparty computation to offer efficient solutions with less interaction and communication overhead.

V. CONCLUSIONS

With the rise of cloud computing and storage, there is rise in the concerns about the security issues of outsourced data. As a result, several constructions have been proposed in recent years to provide with the ability to perform searches and computation on the encrypted data that would greatly reduce the risk of data breaches and insider attacks. Some of the schemes like Functional and property preserving encryption schemes The most well-known of these COED methods, Fully Homomorphic Encryption, can in theory offer a perfect solution, but is extremely inefficient for most real-world situations. Other solutions include Oblivious RAM and multiparty computation and special forms of identity-based encryption, but these are still pretty expensive. The ultimate challenge when it comes to COED is to enable secure computing on huge data sets with controlled leakage.

REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searching on encrypted data. In IEEE Symposium on Security and Privacy, SP'00, pages 44–55, 2000.

[3] S. Kamara, C. Papamanthou, and T. Roeder, Dynamic searchable symmetric encryption. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS'12, pages 965–976, 2012.

[4] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious RAMs. J. ACM, 43(3):431–473, 1996.

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

[6] E. Stefanov and E. Shi. Oblivistore: High performance oblivious cloud storage. In Proceedings of the 2013 IEEE Symposium on Security and Privacy, SP'13, pages 253–267, 2013.

[7] M. Abdalla, M. Bellare, D. Catalano, M. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, H. Shi, Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions, in: Advances in Cryptology, CRYPTO 2005, Springer, Berlin, Heidelberg, 2005, pp. 205–222.

[8] D. Boneh , Amit Sahai, and Brent Waters, Functional Encryption: Definitions and Challenges, Theory of Cryptography Conference, TCC 2011: pp 253-273

[9] Joppe W. Bos, Kristin Lauter, , Michael Naehrig, Private predictive analysis on encrypted medical data, Journal of Biomedical Informatics, Volume 50, August 2014, Pages 234–243

[10] Manish M. Potey C.A. Dhote. Deepak H. Sharma, Homomorphic Encryption for Security of Cloud Data, Procedia Computer Science Volume 79, 2016, Pages 175-181.

[11] Adriana Lopez-Alt, Eran Tromer, Vinod Vaikuntanathan, On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption, Proceeding STOC '12 Proceedings of the forty-fourth annual ACM symposium on Theory of computing Pages 1219-1234

[12] Ivan Damgard, Yuval Ishai, Mikkel Krøigaard, Perfectly Secure Multiparty Computation and the Computational Overhead of Cryptography, Proceeding EUROCRYPT'10 Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques, Pages 445-465.

[13] ElGamal, T, A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31(4), 469–472 (1985)

[14] R. L. Lagendijk, Zekeriya Erkin, and Mauro Barni, Encrypted Signal Processing for Privacy Protection, Conveying the utility of homomorphic encryption and multiparty computation, IEEE Signal Processing Magazine, 2013

[15] David Pointcheval Thomas Johansson , Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE, Conference proceedings, EUROCRYPT 2012

[16] Sander, T., Young, A., Yung, M, Non-interactive Crypto Computing for NC1. In: Proceedings of the 40th Symposium on Foundations of Computer Science (FOCS), New York, NY, USA, October 1999, pp. 554–567. IEEE Computer Society Press, Los Alamitos (1999)

[17] Pallier, P, Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)

[18] Brent Waters, Functional Encryption: Beyond Public Key Cryptography, Identity based encryption workshop 2008.

[19] D. Boneh, E. Kushilevitz, R. Ostrovsky, W.E. Skeith III, Public key encryption that allows PIR queries, in: Advances in Cryptology, CRYPTO 2007, Springer, Berlin, Heidelberg, 2007, pp. 50–67.