Secure Computation for Cloud data Storage

Davood Rezaeipour

Sharif University of Technology, Tehran, Iran

resaeipour@ee.sharif.edu

Abstract

One of the main goals of securing data transmission is focused on the security of cloud data storage. In this paper, we describe several cryptographic techniques which can be used to address the relevant threats and security goals for analyzing cloud computing security. Private semi-trusted clouds, allow researchers to design private clouds by using cryptographic techniques, to protect the semi-trusted ones. Finally, we elaborate on semi-trusted clouds which are related to real-world deployments of cloud resources, and how optimizing cryptographic protocols, would indeed lead to the usage of this certain cloud and therefore practical ways of securing this type of data.

Keywords**:** Multi-party computation, semi-trusted cloud, data security, secure computation**.**

1. Introduction

Cloud computing is predicted to be the next generation of IT enterprise architecture. It's one of the best choices for big data processing and analytics which enables users to remotely store and analyze their data, using shared computing resources. The sharing of resources would in fact reduce the costs of storage for individuals. Cloud services potentials have not been fully unleashed due to users' concerns about security and privacy of their data in the cloud. These concerns are mainly about the cloud operators having the chance of reaching the sensitive data, and therefore reduce the adoptability of cloud computing in many fields, such as the financial industry and governmental agencies. Cloud computing open doors for small and medium scale organizations to perform computations at very low costs. It outsources Software, Infrastructure and Platforms as Services to its clients. The cloud provider and tenants may be untrusted entities who try to tamper with data storage or computation. Preparing to meet and react against these threats, motivate the need of using cryptographic techniques to achieve cloud computing security goals. In summary, the contributions of this paper are as followed:

- Analyzing the differences in various types of cloud data storage and optimizing the cryptographic solutions to achieve secure cloud computing.
- Providing survey and categorizing current cloud-based searchable cryptographic systems.
- Identifying forthcoming challenges in multi-party computation and introducing future research avenues to address them.

While that, the rest of the paper is organized by section 2 introducing the principles of data security in the cloud, which mainly includes trusted, untrusted and semi-trusted clouds. In section 3, we describe the secure multi-party computation techniques which can be used to address the security goals in different cloud deployments. Finally, section 4 will close it with a concluding remark.

2. The Principles of Data Security in the Cloud

The recent study on Cloud computing is mainly focused on the service side, while the data security and trust have not been sufficiently studied yet. Though many techniques on the topics in cloud computing have been investigated in both academics and industries, data security and privacy protection are becoming more important for the future development of cloud computing technology in government, industry, and business. There are many security risks to any sensitive data which involved in cloud computing. Cloud computing is not just a third party data storage, but the need to entrust data protection to a third party cloud provider directs to the protections offered by cryptography in cloud. Actually, when the user data releases to a cloud environment and leaves the protection sphere of owners, then the required guarantees for protecting of data become serious problem.

We believe that the best approach to the user to ensure security of cloud data storage is the use of cryptographic solutions and secure communication issues. These include confidentiality and integrity issues. Confidentiality indicates that all sight sensitive data should be accessible to only "legitimate" receivers and keep secure from any potentially adversarial or untrusted entities. Integrity means that any unauthorized modification of sensitive data is detectable. Thus the outputs of any computation on sensitive data should be consistent with the input data.

Cloud deployment methods and the existent trust between entities are the basic needs to achieve confidentiality and integrity as well. There are three concepts for cloud data storage: Untrusted cloud, trusted cloud and semi-trusted cloud. Untrusted and trusted clouds correspond respectively to the public and private cloud deployment models. Semi-trusted cloud may correspond to the hybrid, public, or private clouds. Untrusted cloud is consistent with a cloud provider who is not trusted to the cloud nodes to maintain the confidentiality or integrity of data. Trusted cloud occurs while the cloud is deployed in an isolated environment from any outside adversaries. But, even in such environment, some nodes may be corrupted due to malicious insiders and lead to violate data and computation integrity. Semi-trusted cloud is consistent with a data owner who attempted to maintain security of sufficient fraction of the cloud resources, but some parts of cloud may be controlled by the untrusted parties. In an attempt to violate the confidentiality or integrity of the data or computations, the corrupted parties are the main threats in order to collecting of additional information through by combining the observations of malicious adversary.

Suppose an agency which is responsible for creating the datasets and maintaining the reference database. The datasets contain metadata computed by the cloud using a set of effective algorithms and user criteria. Now, this agency does not fully trust that its cloud is free of adversaries and wishes to place some trust in its security. Actually, semi-trusted cloud corresponds to ignoring risk of malicious insider taking control of some cloud nodes.

Now, we will describe the variety of cryptographic techniques for securing data in transit, in storage, and in use. We believe these techniques are applicable to achieving secure big data analytics in the cloud and will become an essential part of the big data ecosystem. The future of big data processing depends on close collaboration between cryptography and data science. Homomorphic Encryption, Verifiable Computation and Secure Multi-Party Computation are three cryptographic techniques that can be used to outsource the secure data processing to another entity, trusted to perform the computation correctly. Homomorphic encryption allows functions to be computed over encrypted data while maintaining the confidentiality of data. First scheme developed in 2009. Verifiable computation allows the data owner to check the integrity of the computation, but not trusted to perform the computation correctly. The combination of homomorphic encryption and verifiable computation may be resulted in both confidentiality of the input and output and also integrity of the computation. It enables secure computation over a completely untrusted cloud. Secure Multi-Party Computation is suited to take advantage of the semi-trusted cloud setting, to achieve confidentiality and integrity of the data and computation. It can be used in settings that the different sensitive inputs are held by different parties or in settings where a single client wants to outsource computation on its sensitive input by distributing the computation over multiple compute nodes. First schemes developed in mid 1980's. In this scheme, no single party learns anything about the data, but if many parties are corrupted by an adversary, they can violate confidentiality. Table 1 compares three techniques with respect to their properties.

Table 1. Comparison of cryptographic techniques showing the security guarantees provided and whether computation requires interaction between parties

Cryptographic technique	Adversary type	Confidentiality	Integrity	Interaction between parties
Homomorphic encryption (A)	Malicious	Y	N	N
Verifiable computation (B)	"	N	Y	N
A+B	"	Y	Y	Y
Multi-Party computation	"	Y	Y	Y

We will show that it's possible to create all cryptographic challenges without relying on the trustworthiness of a single party or excluding anyone from participating. We achieve it by using secure multi-party computation. Despite big theoretical progress in recent years, real-world applications for multi-party computation (MPC) are still surprisingly rare. Now, we give a more detailed survey of the secure multi-party computation.

3. Secure Multi-Party Computation

In some cases, data owners want to jointly compute some functions of the collection of their sensitive data. For example, companies wish to predict cyber threats by analyzing the related information from other companies, or hospitals may want to perform medical research on their combined patient data. In these cases, each party wants to get the result of the computation on the data obtained by all parties but without sharing their own sensitive information. Additionally, there may not be a trusted party to whom everyone wishes to allow their information and perform the secure computation. Secure multi-party computation is a computationally efficient approach that allows a client to outsource computation to a group of parties (or cloud providers), assuring that the client's information is prevented from misuse even if some parties are corrupted or cannot fully be trusted. Secure multi-party computation is an area of cryptography that addresses this problem. Its protocols allow parties to perform distributed computation on their private data without the use of a trusted party [19] and exist for any computable function that provide privacy for new applications in which parties currently share their data. In the ideal world, there is a party that everyone trusts with their private inputs and the trusted party would perform distributed computation on outputs. Actually, secure multiparty computation is a powerful cryptographic notion that - in theory - can solve virtually any cryptographic protocol problem. In recent years the technology has been used in practice and holds great promise for future applications.

Cryptographic protocols typically aim to provide security against adversarial behaviors such as *semi- honest adversary* and *malicious adversary*. These adversaries will be done several efforts trying to actively interfere with the computation. Semi-honest adversary follows protocol description and tries to get unauthorized information from the messages he receives. A malicious adversary deviates arbitrary from the protocol in order to get private information or disrupt the protocol.

MPC generally considers these two types of adversaries: semi-honest and malicious. In this manner, we can consider a single adversary who corrupts the subset of the parties, controls their behavior and also their inputs and outputs. We recognize that there are difference between computational security and information-theoretic security. Computational security is based on the assumed hardness of some computational problems, but information-theoretic security is based on the amount of information which adversary can collect.

MPC provides less security guarantee than homomorphic encryption scheme, but more efficient. It is a promising candidate for use in more practical secure cloud computation. MPC on the cloud can be seen in Figure 1. The input holders share the data among the compute nodes, which perform multi-party computation on the shares. The data receiver reconstructs the output. In Figure 1, shaded red nodes are untrusted or adversarial; shaded green nodes are semi-trusted [1].

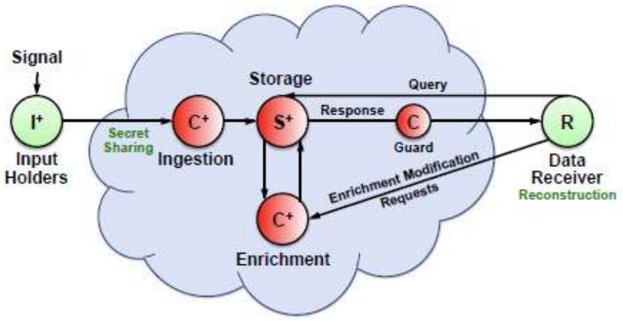

Figure.1. Secure multi-party computation with semi-trusted cloud [I=input node, C=compute nude, S=storage nude, R=result node] X^{+}= one or more nodes of X ($X \in \{I, C, S, R\}$) [1]

MPC technology can be used to implement, for instance, voting, auctions, procurement and benchmarking with better security, in particular without anyone having to reveal his private data to anyone else. What makes secure multi-party computation different than other forms of cryptography is the fact that it treats the participating parties as adversaries.

3.1 Previous Works

Yao [2] introduced the first two-party protocol for computing functions, providing computational security against a semi-honest adversary in 1982. Yao suggested the popular millionaire problem, describing two millionaires interested in knowing which one of them is richer, without revealing their actual worth. Also, millionaire problem with rational players is a unified approach in classical and quantum paradigms. Fairplay [3] was the first compiler for secure two-party computation, using garbled circuits. TASTY [4] combines garbled circuits and homomorphic encryption. Since the development of these early compilers, there have been many implementations of garbled circuits with various optimizations. In the decades since, the two-party problem has been generalized to multi-party computation by Chaum, Crepeau, and Damgard [5] and also Ben-Or, Goldwasser, and Wigderson [6]. FairplayMP [7] extended Fairplay to multiple parties. VIFF [8], SEPIA [9], Choi et al. [20], Sharemind [11], PICCO [12], and Wysteria [13], compile code to secret-sharing based schemes for more than two parties. Secure two-party and multi-party computations have long stood at the center of the theoretical foundations of modern cryptography.

Now, we identify forthcoming challenges in MPC and introduce future research avenues to address them.

3.2 Discussion

Secure multi-party computation allows a set $P = \{p_1, \dots, p_n\}$ of n total parties, which uses a threshold adversary structure $\Pi = \{S : S \subseteq P, |S| \leq t\}$ for some t. This scheme limits the total number of nodes which can be corrupted by an adversary to t out of n total participating parties. It shares the input data among all participating nodes so that no set of fewer than t shares reveals anything about the input data. Then, the nodes make shares of the output which honest parties have enough shares to reconstruct the actual output. Because the cloud nodes have a distinct view of shares, they do not learn the computation output. Multi-party computation based on secret sharing has been more commonly used in production systems. In this scheme, since many standard protocols rely on secret sharing inputs, such a computation would require each party to share its input with the other parties. In a secure MPC, each party possesses some private data, while secret sharing provides a way for one party to spread information on a secret such that all parties together hold full information, yet no single party has all the information.

In order to solving the secure multi-party computation problem, the adversary is specified by a single corruption type (active or passive) and a threshold t on the authorized number of corrupted parties. Goldreich, Micali, and Wigderson [14] proved that, based on cryptographic constraints, the secure multi-party computation is possible if and only if $t < n/2$ parties are actively corrupted. The threshold for passive corruption is $t < n$. In the information-theoretic model, which mutual secure channels between every pair of parties are assumed, Ben-Or, Goldwasser, and Wigderson [6] proved that perfect security is possible if and only if $t < n/3$ for active corruption, and if and only if

$t < n/2$ for passive corruption. Also, some MPC schemes based on the secret-sharing paradigm allow $t < n/3$ malicious corruptions. These results *are* shown in Table 2.

Table 2. The threshold conditions for the secure multi-party computation

Model	Adversary Type	Condition	Reference
Cryptographic	Passive	$t < n$	[14]
	Active	$t < n/2$	[14]
Information-theoretic	Passive	$t < n/2$	[5 , 6]
	Active	$t < n/3$	[6]
MPC scheme based on secret-sharing paradigm	Active	$t < n/3$	[5]

MPC can be used in settings where different sensitive inputs are held by different parties or in settings where a single client wants to outsource computation on its sensitive input by distributing the computation over multiple compute nodes. Although MPC has been studied substantially, building solutions that are practical in terms of computation and communication cost is still a major challenge. There are a number of practical MPC implementations. Most famously, Danish farmers used it in auctions to agree on the price of sugar beets [15]. Among the specific functions of interest in secure multiparty computation, Private Set Intersection (PSI) is probably one of the most strongly motivated by practice [21]. PSI allows n parties to compute the intersection of their datasets without revealing any additional information. The VIFF library [16] runs several MPC protocols and takes 2.1 seconds to evaluate a single AES[1] block [17]. Keller et al. focused on modern secret sharing based MPC protocols and pre-processing of SPDZ and VIFF protocols [26]. Wójcik compared the secure MPC frameworks FRESCO and Bristol SPDZ in terms of infrastructural differences, practicality and performance [22]. Bogdanov et al. developed Sharemind[2], which is secure against t = 1 honest-but-curious corruption. Sharemind is trying to achieve cloud privacy of data computations (of single party) by distributing to multiple cloud servers. Sharemind takes 0.24 seconds to evaluate an AES block [11]. X. Wang et al. proposed a new, constant-round protocol for multi-party computation of boolean circuits that is secure against an arbitrary number of malicious corruptions [23]. Burkhart et al. developed Sepia [9] to be secure against $t < n/2$ honest-but-curious corruptions. This improvement paves the way for new applications of MPC in the area of networking [10]. Gupta proposed the first systematic consideration of Intel's Software Guard Extensions as a platform on which to implement two-party secure function evaluation, facilitating efficient protocols and future work will include the implementation of these protocols and improvements to their efficiency and security properties [24]. Schneider assessed the idea of using one party as a helper (i.e. one party for assisting computation) in the context of secure-multi party computation [25]. Ejgenberg et al. are currently developing an efficient general purpose library called Secure Computation Application Programming Interface (SCAPI[3]) [18], achieving to implement many of the efficient MPC protocols. Finally, Microsoft researchers could enable individuals to share encrypted data through the cloud while giving the owners of that data complete control over specific pieces of information. Users can encrypt and store their data online and share pieces of earmarked information with specific parties.

4. Conclusions

In this paper, we discussed the use of secure multi-party techniques to address cloud computing security goals. Cryptographic techniques and secure communication issues improve users' concerns over security of cloud data storage. These solutions initiate a good event to achieve secure cloud computing in the real world. We showed that MPC is in fact efficient enough to securely create cryptographic challenges. Also, it can be shown that for cryptographic challenges it is sufficient to rely on MPC protocols which are secure against semi-honest adversaries, by verifying honest behavior once a challenge is solved and showing how to combine FHE and MPC to get something much better and practical. Further research is needed to optimize the cryptographic protocols for use in the private cloud which leads to practical solutions for data security in cloud.

[1] AES = Standard Block cipher. The time to compute a one block AES encryption is a commonly used benchmark for testing efficiency of MPC protocols.

[2] Sharemind framework relies on secure multi-party computation, which is secure against $t = 1$ malicious corruptions.

[3] SCAPI (Secure Computation Application Programming Interface) is an open-source general library tailored for secure computation implementations.

References

[1] S. Yakoubov, V. Gadepally,N. Schear, E. Shen, A. Yerukhimovich, A survey of cryptographic approaches to securing big-data analytics in the cloud, In *High Performance Extreme Computing Conference (HPEC)*, 2014 IEEE, pp.1-6, Sept. 2014.

[2] A. C. Yao, Protocols for secure computations (extended abstract), In *Foundations of Computer Science (FOCS),* University of California Berkeley, 1982.

[3] D. Malkhi, N. Nisan, B. Pinkas,Y. Sella, Fairplay - A secure two-party computation system, In *Proc. of the 13th conference on USENIX Security Symposium*, vol. 13, pp.20-20, 2004.

[4] W. Henecka, S. Kogl, A.R. Sadeghi, T. Schneider, I. Wehrenberg, TASTY: Tool for Automating Secure Two-partY computations, In *Proc. of 17th ACM Conference on Computer and Communications Security,* pp. 451-462, CCS 2010.

[5] D. Chaum, C. Crepeau, I. Damgard, Multiparty unconditionally secure protocols (extended abstract), In *Proc. of the 20th Annu. ACM Symposium on Theory of Computing,* pp. 11-19, 1988.

[6] M. Ben-Or, S. Goldwasser, A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract), In *Proc. of the 20th Annu. ACM Symposium on Theory of Computing,* pp. 1-10, 1988.

[7] A. Ben-David, N. Nisan, B. Pinkas, FairplayMP : A system for secure multi-party computation. In *Proc. of the 15th ACM Conference on Computer and Communications Security,* pp. 257-266, CCS 2008.

[8] I. Damgard, M. Geisler, M. Krigaard, J.B. Nielsen, Asynchronous multiparty computation: Theory and implementation, In *Proc. of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09*, pp.160-179, 2009.

[9] M. Burkhart, M. Strasser, D. Many, X. Dimitropoulos, SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics, In *Proc. of 19th USENIX Conference on Security*, pp. 15-15 , 2010.

[10] M.V. Maltitz, S. Smarzly, H. Kinkelin, G. Carle, A Management Framework for Secure Multiparty Computation in Dynamic Environments, *arXiv: 1804.03918v1*, April 2018.

[11] D. Bogdanov, S. Laur, J. Willemson, Sharemind: A framework for fast privacy-preserving computations, In *Proc. of the 13th European Symposium on Research in Computer Security - ESORICS'08*, pp. 192-206, 2008.

[12] Y. Zhang, A. Steele, M. Blanton, PICCO: A general-purpose compiler for private distributed computation, In *Proc. of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 813-826, 2013.

[13] A. Rastogi, M. A. Hammer, M. Hicks , Wysteria: A programming language for generic, mixed-mode multiparty computations, In *Proc. of the 2014 IEEE Symposium on Security and Privacy*, pp. 655-670, 2014.

[14] O. Goldreich, S. Micali, A. Wigderson , How to play any mental game, In *Proc. of the 19th annual ACM symposium on Theory of computing*, pp. 218-229, 1987.

[15] P. Bogetoft, D. Christensen, I. Damgard, M. Geisler, T. Jakobsen, M. Krøigaard, J.D. Nielsen, J.B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, T. Toft, Secure multiparty computation goes live, In *Financial Cryptography and Data Security*, pp.325-343, 2009.

[16] - , VIFF: the Virtual Ideal Functionality Framework, http://viff.dk/ , Accessed 01/08/2016.

[17] I. Damgard, M. Keller, Secure multiparty AES, In *Proc. of the 14th international conference on Financial Cryptography and Data Security*, pp. 367-374, 2010.

[18] Y. Ejgenberg, M. Farbstein, M. Levy, Y. Lindell, SCAPI: The secure computation application programming interface, *IACR Cryptology ePrint Archive*, 2013.

[19] A. Acar, Z.B. Celik, H. Aksu, A.S. Uluagac, P. McDaniel, Achieving Secure and Differentially Private Computations in Multiparty settings, *arXiv: 1707.01871v1*, July 2017.

[20] S. G. Choi, K. W. Hwang, J. Katz, T. Malkin, D. Rubenstein, Secure multi-party computation of boolean circuits with applications to privacy in on-line marketplaces, In *Proc. of the 12th Conference on Topics in Cryptology*, pp. 416-432, 2012.

[21] V. Kolesnikov, N. Matania, B. Pinkas, M. Rosulek, N. Trieu, Practical Multi-party Private Set Intersection from Symmetric-Key Techniques, In *Proc. of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1257-1272, 2017.

[22] J. Wójcik, Practical Assessment of Secure Multiparty Computation Frameworks, The chair of Network Architectures and Services, Technische Universität München, https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2018-03-1/NET-2018-03-1_03.pdf , Germany, 2018.

[23] X. Wang, S. Ranellucci, J. Katz, Global-Scale Secure Multiparty Computation, In *Proc. of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 39-56, 2017.

[24] D. Gupta, Practical and Deployable Secure Multi-Party Computation, PhD Thesis, Yale University, 2017.

[25] J. Schneider, Secure Multi-Party Computation with a Helper, *arXiv: 1508.07690v5*, October 2018.

[26] M. Keller, P. Scholl, N.P. Smart, An architecture for practical actively secure MPC with dishonest majority, In *Proc. of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 549-560, 2013.