Encryption and privacy concerns in Cloud Computing

JYOTI
BRCM ,Bahal, Hisar, India
Email: Jyotipanghal10@gmail.com

Abstract— ***Cryptographic system ensures privacy of data/operations while being processed at unsecure servers.It has been indispensable tool for computer security.Readiness of Crtyptography towards new cloud computing is still doubtful.***

Few directions where cryptography techniques being pursued are Functional encryption, Server aided Multiparty Computation, Fully Homomorphic Encryption,,Verifiable Computation.These techniques solve Cloud piracy issues at various levels.

Different delivery methods are used to state privacy requirements for Cloud offering.Various crytographics techniques pursued are cross verified to challenged by researchers and shown that they dont cater to blanket cover these privacy requirements.

We need identify connection among various isolated techniques.This may lead to more insights in underpinnings of computational privacy.

I. INTRODUCTION

Cloud computing came out of age. As it always happens, security is being an afterthought. Before it is too late, its time for us to think through the tools available for us for secure cloud offerings.

To best of our knowledge, the cryptographic techniques for solving cloud computational privacy problems are micro- scopic, in the sense that, the protocols, schemes, mechanisms being devised solve a discrete subset of problems. We provide a panoramic view of privacy problems in various cloud delivery methods and current cryptographic technology landscape.

We emphasize the need for further generalization of differ- ent approaches and formalize the theory behind Computational Privacy for Cloud Computing.

A. Prior Work

With similar goals as stated in our work, a paper stating the impossibility of cryptography alone for solving privacy preserving Cloud Computing has appeared [2]. Their central idea talks about the impossibility of Fully Ho mo mo rp h i c Encryption (FHE) alone for Cloud privacy. Their classification hierarchy of Cloud Computing is not standard model and has few shortcomings as we would discuss duly. We state the security and privacy issues from standard Cloud Computing definitions and discuss the challenges involved not just for FHE but also for many other techniques.

B. Outline

The rest of the paper is organized as follows. Section 2 contains a quick introduction to Cloud Computing delivery methods and deployment models. Section 3 contains the se- curity and privacy requirements of Cloud delivery methods. Section 4 contains a guided tour of various cryptographic tools available, problems they try to solve and their challenges for adopting to Cloud setup. Section 5 contains conclusions.

II. CLOUD COMPUTING

Cloud computing has been standardized now. The princi- ples defining the essential characteristics, delivery methods and deployment models are now well laid and widely accepted[1].

A. Delivery Methods

The three delivery methods of Cloud Computing are

1) Software-As-a-Service (SaaS): In this method the user does not purchase software, but rather rents it for use on a subscription or pay-per-use model (an operational expense, known as OpEx). In some cases, the service is free for limited use. Example: Gmail , Google Drive, DropBox etc.

2) Platform-As-a-Service (PaaS): In this method, the ser- vice provider offers a development environment to application developers, who develop applications and offer those services through the providers platform. Example: Google Gears, Microsoft Azure

3) Infrastructure-As-a-Service (IaaS): In this method, the service provider offers compute, storage and networking ca- pabilities to the user. The user would be able to run any arbitrary software of his own including operating systems etc. The physical infrastructure is handled by service provider at a remote place and virtual abstractions are given to the user. Example: Amazon Web Services, Google's Compute Engine.

B. Deployment Models

The deployment model for the discussion throughout this paper would be Public Clouds (and also Hybrid Clouds). In Public Cloud deployments Cloud platform cannot be relied upon as the cloud infrastructure is run at

service provider premises and open for public use. In Hybrid Clouds too part of the cloud infrastructure is run at service provider. Whereas in Private Cloud deployments the platform can be trusted since its completely within users premises

III. SECURITY AND PRIVACY OF CLOUD

An excellent and detailed coverage on the security and pri- vacy requirements for Cloud are covered in [3]. Simplistically said Cloud Computing is about a user giving away data to server for carrying out some computation. We classify them based on few simple questions

- Is the data safe from unauthorized access (privacy) ?
- Is the data (or code) tampered (integrity) ?
- Is the result correct (verifiability) ?

A. *Privacy :*

In general the term confidentiality is used for limiting access to data and privacy for users, we use the term privacy to represent both in this paper. We further classify privacy issues into data and operational

1*) Data Privacy:* refers the privacy issues related to entire user data outsourced to Cloud. It is to be noted that privacy issues like data-in-transit and data-at-rest are solved using traditional cryptographic techniques and are not part of current discussion. Further requirements would be

- All the inputs, outputs of the computations being performed at Cloud server should be encrypted
- It should be possible to enforce Access Control over encrypted data as the users would have different levels of trust with different types of users.
- The intermediate results should be leak proof and data flow paths should be protected.
- Accidental data remanence by delete, erase operations should not leak any details.

Intuitively, all of the above requirements can be met, if we could carry out computations over encrypted data.

2) Operational Privacy: In SaaS delivery method often the operations being executed on users data is pre-defined by Cloud Service Provider. So achieving the privacy of such operations itself may not be desired. However in PaaS and IaaS delivery methods the user would run set of applications defined by her and achieving the privacy of such operations (or applications) may be desirable.

B) Integrity: The property is inspired from classic com-munication security. The unit of communication is packet for which integrity can be defined and verified. In Cloud Computing setup, this gets tricky, since the unit of computation cannot be defined. Even if we did it would be highly inefficient in real time. Integrity holds good though in long term Cloud storage solutions, note that our current discussion is only for computations. One might argue that there is need to ensure integrity of operations (i.e tamper proofing code) in PaaS and IaaS models. As long as the set of operations are stored on the drive of remote server, measures taken to ensure integrity of long term storage of data can be applied. At run time when the operations are being executed, verification of such integrity, would not be of much use. Either they would be highly inefficient or an adversary compromising the Cloud platform itself can bypass such measures. So for this reason we strongly emphasize the need for verifiability of the computation.

C) Verifiability: As we have seen that integrity has not much of relevance in this current context. Verifiability (or Provenance) of the computation is very important. User should be able to verify that the results of the computations are in fact correct. In other way the Cloud server should be able to prove the validity of the results. We summarize these requirements in the below table

TABLE I. SECURITY AND PRIVACY REQUIREMENTS

Requirement	SaaS	PaaS	IaaS
Data privacy	Yes	yes	yes
Operational privacy	-	yes	yes
Verifiability	Yes	yes	yes

4) Adversaries: Multiple real world adversaries exist, the Cloud platform itself, somebody compromising Cloud plat- form, neighbors sharing the platform but all of them can be modeled as single adversary. Such differentiation is not important if the privacy and verifiability measures are in place.

IV. CRYPTO TECHNIQUES AND C HALLENG ES

Many variants of computational privacy problems are being formalized cryptographically and solutions are being proposed. Success has been achieved in various degrees in many of them. We give a whirlwind tour of those techniques here and discuss the challenges associated with each of these for adopting to Cloud Computing.

A. Fully Homomorphic Encryption (FHE)

In addition to the Keygen, Encrypt, Decrypt methods of Public Key Encryption (PKE) schemes, these schemes provide an additional algorithm Evaluate. Such algorithm allows computations over encrypted data, based on mathematical property called homomorphism that performs basic operations like addition and multiplication on cipher text. Recent breakthroughs in FHE [4] has got them wide attention and they are often thought to be cryptographic holy grail. They are currently highly inefficient and not practical [10].

Using these schemes, in any of the Cloud deployment methods, a client encrypts the data using the Public Key and outsources it to a Cloud server for any

computations. The server would perform computations over the encrypted data using the Evaluate method and the Public Key and returns the results in encrypted form. The client then decrypts the results locally using Private Key.

FHE schemes are two party (one client and server) models. Xiao et all proposed protocols for Multi User systems [5], that are based on symmetric Homomorphic Encryption scheme that could evaluate functions only on polynomials. Their protocols are tightly coupled with their scheme. Such limitation to two party model, make them suitable for outsourcing intensive sci- entific computations but not for commercial Cloud applications yet.

Similar concern was raised by Dijk, Juels in their contradictory paper [1]. They claim multi client applications would be impossible due to additional functionality needed like access control, re-encryption etc. We believe such additional functionality is purely application of FHE. [2]

These schemes are safe only in semi-honest adversarial model where the Cloud service provider is assumed to be honest in performing the computations but are curious to get more information than they are ought to know. But in real world, we cannot assume any degree of honesty, some adversary compromising the Cloud platform itself might turn the provider malicious to corrupt the data and/or computations. For this reason, Verifiability of computation is very important, the current proposed techniques that suit in Cloud Computing setup are still nascent [7] [8].

FHE schemes are malleable [3] by design. For this reason, they would be prone to adaptive chosen cipher text (CCA2) attack, in which an attacker gradually reveals the decryption key or plain text itself. This is also equivalent to, informally, an adversary being able to distinguish the cipher text based on the message they encrypt. In practice, malleability is avoided using padding methods like OAEP or PKCS1

Also popular PKE schemes like RSA in their basic form are deterministic in nature. It means encrypting the same message any number of times would yield same cipher text. This would leak information to an adversary if the data contains repeated patterns. In practice, the encryption process is made proba- bilistic using padding methods. So the choice of PKE schemes underlying FHE schemes should be inherently probabilistic in nature, else their ability to compute over encrypted data might be lost due to padding.

Little is known yet on how FHE schemes can ensure opera- tional privacy. So far only evaluation of encrypted polynomials is possible[9]. If FHE can guarantee operational privacy then it would contradict very important results on program obfuscation [10]. Few researchers even proved that achieving multi user computational privacy implies program obfuscation [1]. So it is open problem still if FHE can guarantee generic operational privacy in Cloud setup.

B. Server aided Secure Multiparty Computation

Secure Multiparty Computation (SMC) solves the problem of evaluating a function jointly by multiple parties on their private inputs [11]. In their basic form these techniques have been developed for few distrustful parties to evaluate a com- mon function over their private inputs. These protocols are highly interactive in nature. Also no assumptions are made on computational resources available with the parties. All the parties would carry out same amount of work which is contrary to Cloud Computing setting.

To adapt these techniques for an asymmetric setting like Cloud Computing where the server has massive amounts of computing power relative to the users, Server aided SMC techniques have been proposed [12]

Fundamentally SMC has been proposed to carry out the computations among untrusted parties. Where as in Cloud Computing model trusted parties need to carry out computation in the presence of an untrusted server. Even in a multiuser scenario, the user trusts (with various degrees) rest of the users except for the server. For example, If Patient Health Records processing is out sourced to a Cloud server, the patient would trust and share parts of the information with Doctors, Insurance Companies, Drug Researchers etc with various degrees but may not trust the remote Cloud server itself where the processing is being carried out.

SMC also does not make any assumptions on computing resources available with participants. But where as in Cloud Computing setup, server has massive computing power com- pared to the user.

SMC are highly interactive protocols that expect the users to be always online, where as in Cloud model this expectation may not be reasonable. When just two users are involved who don't trust each other, these techniques can be adapted for few applications in Cloud setup. But when the number of users grows in SMC the protocol interactions visually represent more of mesh but where as in Cloud Computing they represent a hub-spoke model, the hub being the server.

Also set of literature exists for achieving multi party computation using threshold homomorphic encryption [13],[14] and also multi-key homomorphic encryption [15]. These techniques require few of the users to collaborate interactively to decrypt the final result, which is not reasonable to assume especially in the Cloud Computing kind of setup.

So for the reasons stated above adapting SMC or its variants for Cloud Computing setup may not be of much help. Also there is literature around realizing SMC protocols using FHE. It would be interesting though to see if FHE can be realized using SMC.

C. Functional Encryption

Traditional encryption schemes are all-or-nothing meaning either the cipher text can be decrypted in its entirety or nothing can be done. But often applications would need users to have access control over the data, that could reveal parts of the data based on predefined privileges. Interestingly below are few schemes that allow to do the same

- Identity based encryption
- Attribute based encryption
- Predicate-based encryption

In all of the above techniques, the data owner encrypts the data using public key and also predefines granular access privileges for the rest of the users to access it. Users would then get secret keys from a trusted key server and then decrypt parts of the encrypted data based on their assigned privileges. Such property is very important when different levels of access control needs to be enforced on the encrypted data. But by design they do not provide Output Privacy required in Cloud Computing set up.

Generalization of the above techniques has been formalized as Functional Encryption[16]. Such generalization is an important step towards a unified theory for computational privacy. Interestingly its relations with FHE has been studied [17] and connections have been established. This gives us a hope that Functional encryption can be further generalized with additional restrictions for output privacy.

D Instance hiding (IH)

If a user wants to outsource the computation of a function for a particular input x (instance). She transforms the input x to an encrypted input y (thus hides it) in such a way that the server cannot infer x from y and sends to the server. The server computes the function on y and returns the result. The user then transforms the result f(y) back to the value of f (x). These techniques are called Instance Hiding techniques [18] as they hide the actual inputs from the server. The functions that can be evaluated this way are called encryptable functions.

Few protocols were also proposed to achieve operational privacy [19] using these techniques.

Prima Facie these techniques look they can be adapted for Cloud setting. But it has been proved that not all functions are encryptable, this means not many functions can be evaluated when the real input instances are hidden from the server.

If there aren't many encryptable functions then the results look contradicting with recent breakthroughs of FHE schemes. FHE schemes aim to perform generically all functions by computing fundamental operations like add, multi on trans- formed inputs. Of course there is no formal analysis done on connections between both of them.

E . Superimposing encrypted data

Although not so popular, its been proposed that efficient encryption of data is possible using time-reversal transforma- tions [20]. Further using this technique, the possibilities of processing over encrypted data has been explored using super imposing such encrypted data [21].

Interestingly these techniques are inspired from principles of Physics. Quoting verbatim from their work

The fact that two ciphertexts can be superimposedwhile each retains its original pattern is analogous to the superposition of waves

Not much analysis is available on these techniques. We admit our own limited knowledge in this area to do thorough analysis. We mention this, so that the community may find it useful to know an obscure technique.

F. Hardware approaches

Tamper Proof Hardware approaches have been proposed to process encrypted data. In short, the devices have the decryption key built in, all the inputs are fed to the device in encrypted form, the processing is done by decrypting them and the results are re-encrypted again. Few approaches proposed could achieve operational privacy by running encrypted programs [22]. Few techniques were even successful in evading few types of side channel attacks [23] Hardwiring of the decryption key is risky proposition; compromise of the key through any side channel attacks would render the device useless and compromised forever. Even if rekeying was possible, it would be costly affair.

The success of Cloud Computing can be attributed to optimum utilization of underlying hardware resources using Virtualization. Abstraction of a virtual machine gives the flexibility to run on programs on shared resources. So adopting techniques that require specialized hardware would lose such abstraction and flexibility.

Also manufacturing specialized hardware would shoot up the prices thus defeating the purpose of moving to Cloud. Even if the additional cost is amortized over a period of time, its highly unreasonable to assume the users decryption key to be residing in the datacenter of Cloud server for which user has no control.

G. Specialized Operations

1) Proxy re encryption: techniques allows to translate a cipher text encrypted under one key to cipher text encrypted under another key without every decrypting it, provided some additional information [24], [25]. These techniques are used in distributed secure storage.

2) Searchable encryption: techniques allow performing search over encrypted data [26], [27]. These techniques have been improved and implemented in MIT's cryptdb project [28].

3) SQL-Aware encryption: is a strategy rather than a technique in itself. Its based on the fact that all SQL Queries are made up of well defined primitive operations like add, equality, order check etc. So a collection of encryption schemes that allow these operations have been engineered into an RDBMS application. This made possible to execute SQL-like queries on encrypted databases [28].

These specialized techniques cater to small subset of functionality that can achieve. Finding connections and realizations of these specific techniques with much more generic techniques like FHE or FE might give us insights into possible efficient solutions.

V.CONCLUSION

Success has been achieved in various techniques at certain level in many previous sections discussed above.Several open issues have been found.. Computing is solving these separated techniques day by day. Industries are converting these half design solutions into products. Strong need of further generalization and formalization of all isolated techniques behind computational privacy for cloud computing has arised.

ACKNOWLEDGMENT

I would like to thank GARS infotech for supporting this work. I would like to thank my research advisor Dr Amit Kumar at BRCM-Bahal for all the insightful discussions and brainstorming. I would like to thank Mr.Sudesh Kumar for all the opportunities he created for me to learn cryptography and Cloud security in detail.

REFERENCES

[1] P. Mell and T. Grance, "The nist definition of cloud computing, special publication 800-145," US Department of Commerce, Gaithersburg, MD, 2011.

[2] M. Van Dijk and A. Juels, "On the impossibility of cryptography alone for privacy-preserving cloud computing," in Proceedings of the 5th USENIX conference on Hot topics in security. USENIX Association, 2010, pp. 1–8.

[3] T. Mather, S. Kumaraswamy, and S. Latif, Cloud security and privacy: an enterprise perspective on risks and compliance. O'Reilly Media, Incorporated, 2009.

[4] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. disserta- tion, Stanford University, 2009.

[5] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang, "On the (im) possibility of obfuscating programs," in Advances in CryptologyCRYPTO 2001. Springer, 2001, pp. 1–18.

[6] V. Vaikuntanathan, "Computing blindfolded: N ew developments in fully homomorphic encryption," in Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on. IEEE, 2011, pp. 5–16.

[7] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable com- puting: Outsourcing computation to untrusted workers," in Advances in Cryptology–CRYPTO 2010. Springer, 2010, pp. 465–482.

[8] S. G. Choi, J. Katz, R. Kumaresan, and C. Cid, "Multi-user non- interactive verifiable computation."

[9] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in Proceedings of the 3rd ACM workshop on Cloud computing security workshop. ACM, 2011, pp. 113–124.

[10] L. Xiao, O. Bastani, and I.-L. Yen, "An efficient homomorphic encryp- tion protocol for multi-user systems."

[11] O. Goldreich, "Secure multi-party computation," Manuscript. Prelimi- nary version, 1998.

[12] S. Kamara, P. Mohassel, and M. Raykova, "Outsourcing multi-party computation," h ttp://eprint. iacr. org/2011/272. pdf, 2011.

[13] R. Cramer, I. Damga˚rd, and J. B. Nielsen, Multiparty computation from threshold homomorphic encryption. Springer, 2001.

[14] S. Myers, M. Sergi, and A. Shelat, "Threshold fully homomorphic encryption and secure computation," eprint, vol. 454, p. 2011, 2011.

[15] A. Lo´ pez-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in Proceedings of the 44th symposium on Theory of Computing. ACM, 2012, pp. 1219–1234.

[16] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in Theory of Cryptography. Springer, 2011, pp. 253–273.

[17] J. Alwen, R. Gennaro, and D. Gordon, "On the relationship between functional encryption and fully homomorphic encryption."

[18] M. Abadi, J. Feigenbaum, and J. Kilian, "On hiding information from an oracle," Journal of Computer and System Sciences, vol. 39, no. 1, pp. 21–50, 1989.

[19] M. Abadi and J. Feigenbaum, "Secure circuit evalu ation," Journal of Cryptology, vol. 2, no. 1, pp. 1–12, 1990.

[20] K. Yu and T. L. Yu, "Data encryption based upon time reversal transformations," The Computer Journal, vol. 32, no. 3, pp. 241–245, 1989.

[21] K. W. Yu and T. L. Yu, "Superimposing encrypted data," Communica- tions of the ACM, vol. 34, no. 2, pp. 48–54, 1991.

[22] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," Journal of the ACM (JACM), vol. 43, no. 3, pp. 431–473, 1996.

[23] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in Advances in Cryptology-CRYPTO 2003. Springer,

2003, pp. 463–481.

[24] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxycryptography,"in Advances in CryptologyEUROCRYPT'98. Springer, 1998, pp. 127–144.

[25] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re- encryption," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 185– 194.

[26] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44–55.

[27] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryp- tion revisited: Consistency properties, relation to anonymous ibe, and extensions," Journal of Cryptology, vol. 21, no. 3, pp. 350–391, 2008.

[28] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: protecting confidentiality with encrypted query processing," in Pro- ceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM, 2011, pp. 85–100