INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY (IJIT)

Volume 1, Issue 1,
January- June (2014), pp. 10-16

IJIT
© I A E M E

PROTECTION OF DATA USING LINEAR PROGRAMMING AND FHE TECHNIQUE IN CLOUD COMPUTING

G.Ashok kumar[1], P.Srinivasulu[2]

[1]Pursuing M. Tech (CSE), [2] Professor&HOD (CSE)
[1]QIS college of Engineering and technology, Vengamukkalapalem, ongole, Andhra Pradesh, India.
[2]QIS college of Engineering and technology, Vengamukkalapalem, ongole, Andhra Pradesh, India.

ABSTRACT

Cloud computing has good services like virtualization. Virtualization provides the unlimited computational resources. Cloud computing provides robust design with low cost. Different security constraints are satisfied in outsourcing with the implementation of new encryption standards. These above services give the reliable solution in secure transmission. Previous systems cloud environment enables the computational resources are limited whenever access the resources in outsourcing. These resources utilization are pay per use manner here. Previous servers have processing storage, memory levels are less. There is no possibility for encryption complete content. Some content available as a plain text, remaining content available as a cipher text. This two type's format content starts the transmission. Attackers are entering automatically leakage of data problems generate here. That's why this type of network comes under insecure. It can deliver the incorrect data in destination. Users are not satisfies with the help of these services. Security is the primary obstacle that prevents the wide adoption of this promising computing model, especially for customers when their confidential data are consumed and produced during the computation.

The above limitations are overcome using the linear programming in cloud computing. These types of techniques are providing good secure network and optimization solution. User is ready for transfer the large file to another user. Here large file assume as a large problem. Using linear programming large files divide into sub parts using decomposition. Transformation techniques start the allocation of decomposed parts in different servers. Different servers provide the perfect infrastructure for encryption with sufficient computational resources. Before starts the outsourcing total content is encrypted in client side. After deliver the content verifies the proof. Proof it is matched with server proof then performs the decryption. It's delivers as a correct data.

International Journal of Information Technology (IJIT), Volume 1, Number 1, January- June (2014)

Keywords: linear programming, cloud computing, encryption techniques, outsourcing techniques

I. INTRODUCTION

Cloud computing is a comprehensive Internet-based computing solution. The flexibility of cloud computing is a function of the allocation of resources on demand. While a traditional computer setup requires you to be in the same location as your data storage device, the cloud takes away that step .It makes possible for us to access our information from anywhere at any time. The main concerns voiced by those moving to the cloud are security and privacy. The companies supplying cloud computing services know this and understand that without reliable security, their businesses will collapse. So security and privacy are high priorities for all cloud computing entities. The main focus of this paper is not only to protect confidential data from various malicious modifications but also to give a proof that the computed result is correct as per request. For this, linear programming computations are decomposed into public LP solvers. Here the original LP problem is converted into an arbitrary problem which helps to protect confidential information's stored in the cloud and also facilitates the users with an efficient result verification mechanism. One fundamental advantage of the cloud paradigm is computation outsourcing; On the one hand, the outsourced computation workloads often contain sensitive information, such as the business financial records, proprietary research data, or personally identifiable health information etc. However, ordinary data encryption techniques in essence prevent cloud from performing any meaningful operation of the underlying plaintext data, making the computation over encrypted data a very hard problem. On the other hand, the operational details inside the cloud are not transparent enough to customers. As a result, there do exist various motivations for cloud server to behave unfaithfully and to return incorrect results. To protect the sensitive input and output information of the workloads and to validate the integrity of the computation result. Based on Yao's garbled circuits and Gentry's breakthrough work on fully homomorphism encryption (FHE) scheme, a general result of secure computation outsourcing has been shown viable in theory, where the computation is represented by an encrypted combinational Boolean circuit that allows to be evaluated with encrypted private inputs. due to the extremely high complexity of FHE operation as well as the pessimistic circuit sizes that cannot be handled in practice when constructing original and encrypted circuits. Linear programming is an algorithmic and computational tool which captures the first order effects of various system parameters that should be optimized, and is essential to engineering optimization. As LP computations need enough computational power & involve confidential data, in this paper a mechanism is introduced to decompose the LP computation outsourcing into public LP solvers running on the cloud and private LP parameters which is owned by the customer. The first step in this LP problem solving mechanism is to formulate the private data of the customer into some set of vectors and matrices. This representation helps us to deploy some set of privacy preserving problem transformation techniques.

In order to validate the computational result, the fact that the result is from cloud server solving the transferred LP problem can be utilized and along with that the duality theorem, together with the piece wise construction of auxiliary LP problem is used for devising some necessary conditions that the correct result should satisfy.

II. THE PROPOSED SYSTEM ARCHITECTURE

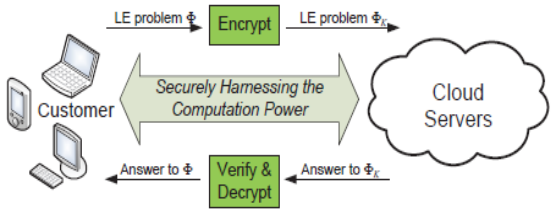

Fig. 1: Architecture of secure outsourcing problems of large-scale systems of linear equations in Cloud Computing

Fully homomorphic encryption (FHE) scheme, a general result of secure computation outsourcing has been shown viable in theory, where the computation is represented by an encrypted combinational Boolean circuit that allows to be evaluated with encryption of their input to produce an encryption of their output. We also investigate duality theorem and derive a set of necessary and sufficient condition for result verification. Such a cheating resilience design can be bundled in the overall mechanism with close-to-zero additional overhead. Both security analysis and experiment results demonstrate the immediate practicality of the proposed mechanism.

Duality in linear programming is essentially a unifying theory that develops the relationships between given linear program and another related linear program stated in terms of variables with this shadow-price interpretation. The importance of duality is twofold. First, fully understanding the shadow-price interpretation of the optimal simplex multipliers can prove very useful in understanding the implications of a particular linear-programming model. Second, it is often possible to solve the related linear program with the shadow prices as the variables in place of, or in conjunction with, the original linear program, thereby taking advantage of some computational efficiency. The importance of duality for computational procedures will become more apparent in later chapters on network-flow problems and large-scale systems.

III. AIM OF THE SYSTEM

To enable secure and practical outsourcing of LP under the aforementioned model, our mechanism design should achieve the following security and performance guarantees.

- **Correctness**: Any cloud server that faithfully follows the mechanism must produce an output that can be decrypted and verified successfully by the customer.
- **Soundness**: No cloud server can generate an incorrect output that can be decrypted and verified successfully by the customer with non-negligible probability.
- **Input/output privacy**: No sensitive information from the customer's private data can be derived by the cloud server during performing the LP computation.
- **Efficiency**: The local computations done by customer should be substantially less than solving the original LP on his own. The computation burden on the cloud server should be within the comparable time complexity of existing practical algorithms solving LP problems.

IV. ALGORITHM USED

The general working procedure is adopted from a generic approach proposed by R. Gennaro, C. Gentry,and B. Parno while the instantiation in this paper is completely different. According to this approach, the process on cloud server can be represented by algorithm ProofGen and the process on customer can be organized into three algorithms (KeyGen, ProbEnc, ResultDec).

- KeyGen (1k) → {K}. This is a randomized key generation algorithm which takes a system security parameter k, and returns a secret key K that is used later by customer to encrypt the target LP problem.
- ProbEnc (K, _) → {_K}. This algorithm encrypts the input tuple _ into _K with the secret key K. According to problem transformation, the encrypted input _K has the same form as _, and thus defines the problem to be solved in the cloud.
- ProofGen (_K) → {(y, □)}. This algorithm augments a generic solver that solves the problem _K to produce both the output y and a proof □. The output y later Decrypts to x, and □ is used later by the customer to verify the correctness of y or x.
- ResultDec (K, _, y, □) → {x,□}. This algorithm may choose to verify either y or x via the proof □. In any case, a correct output x is produced by decrypting y using the secret K. The algorithm outputs □ when the validation fails, indicating the cloud server was not performing the computation faithfully.

The proposed algorithm provides one-time-pad types of flexibility where we should never use the same secret key K to two different problems. Overall, the basic techniques would choose a secret key K = (Q, λ, θ) and encrypt the input tuple γ into γ_k = (A′ ,B′ , b′ , θc), which gives reasonable strength of problem input hiding. Also, these techniques are clearly correct in the sense that solving γ_k would give the same optimal solution as solving γ. However, it also implies that although input privacy is achieved, there is no output privacy. Essentially, it shows that although one can change the constraints to a completely different form, it is not necessary the feasible region defined by the constraints will change. Therefore, any secure linear programming mechanism must be able to not only encrypt the constraints but also to encrypt the feasible region defined by the constraints.

V. RESULT OF FULLY HOMOMORPHISM ENCRYPTION: SECURITY

Homomorphism encryption schemes that are not semantically secured, like basic RSA, may also have stronger attacks on their one-wayness. A homomorphic encryption (HE) scheme encrypts data in such a way that computations can be performed on the encrypted data without knowing the secret key. fully homomorphic encryption schemes is based on creating a function to perform two atomic operations which will allow the user to build any kind of circuit. Effectively, any circuit can be built with two atomic functions, namely addition + and multiplication *. Therefore, to evaluate any circuit, we are only required to be able to add and multiply over F2 two encrypted bits. Gentry used a simple model. Gentry defined the two functions f+ and f* which are equivalent to decrypting both encrypted bits, adding or multiplying such decrypted bits and then encrypting the resulting bits. Hence, it can remove the first encryption securely to perform the addition or the multiplication. Using such a technique, Gentry simplified the quest of constructing a fully homomorphic encryption that can evaluate any circuit on encrypted data by finding an encryption system that can evaluate

only some short circuits, namely f+ and f*. that RSA supports multiplications over encrypted data, i.e., given the encryptions of two messages anyone can compute the encryption of their product . It turns out that many public-key encryption schemes are homomorphic including multiplicatively homomorphic to do addition, multiplications and XOR over encrypted data for a long time and even being able to perform these simple operations has been tremendously useful

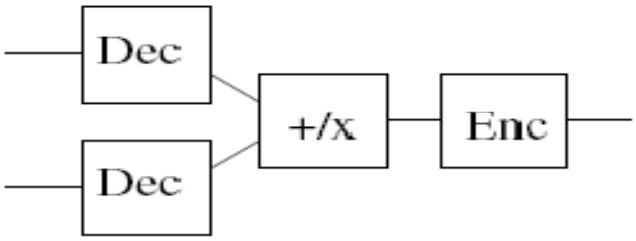

Fig 2. Fully homomorphic encryption model

Protecting FHE with Verifiable Computation cloud will not be able to modify the computation circuit without be detected. As a result, our attack will be unsuccessful. However, we argue that in order to use FHE in those models, one must use verifiable computation all the time. Although the cost of verification is low, to generate the minimum circuit and to homomorphically modify it is costly. Thus, whether this technique can be used in practice is doubtful.

VI. ESTIMATING PERFORMANCE

Customer side computation overhead consists of key generation, problem encryption operation, and result verification, which corresponds to the three algorithms KeyGen, ProbEnc, and ResultDec, respectively. Because KeyGen and Result-Dec only require a set of random matrix generation as well as vector-vector and matrix-vector multiplication, the computation complexity of these two algorithms are upper bounded via O(n2). Thus, it is straight-forward that the most time consuming operations are the matrix-matrix multiplications in problem encryption algorithm ProbEnc. Since $m \leq n$, the time complexity for the customer local computation is thus asymptotically the same as matrix-matrix multiplication.

For cloud server, its only computation overhead is to solve the encrypted LP problem γ_k as well as generating the result proof ∂, both of which correspond to the algorithm ProofGen. If the encrypted LP problem γ_k belongs to normal case, cloud server just solves it with the dual optimal solution as the result proof ∂, which is usually readily available in the current LP solving algorithms and incurs no additional cost for cloud. If the encrypted problem γ_k does not have an optimal solution, additional auxiliary LP problems can be solved to provide a proof. Thus, in all the cases, the computation complexity of the cloud server is asymptotically the same as to solve a normal LP problem, which usually requires more than O (n3) time.

VII. CONCLUSION & FUTURE GOAL

In this paper, Customers to secretly transform the original LP into some arbitrary one while protecting sensitive input/output information. Fully homomorphic encryption (FHE) scheme, a general result of secure computation outsourcing has been shown viable in theory, where the computation is represented by an encrypted combinational Boolean circuit that allows to be evaluated with encrypted private inputs we also investigate duality theorem and derive a set of necessary and sufficient condition for result verification. we show that for any problem γ and its encrypted version γ_k, solution μ computed by honest cloud server will always be verified successfully. This follows directly from the duality theorem of linear programming. Therefore all conditions derived from duality theorem and auxiliary LP problem construction for result verification is necessary and sufficient. Similar to correctness argument, the soundness of the proposed mechanism follows from the facts that the LP problem γ and γ_k are equivalent to each other through affine mapping, and all the conditions thereafter for result verification are necessary and sufficient. In near future a goal is set to work around some interesting concepts such as to devise robust algorithms to achieve numerical stability; to explore the sparsity structure of problem for further efficiency improvement; to establish formal security framework; and also to extend our result to non-linear programming computation outsourcing in cloud.

REFERENCES

[1] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. H. Spafford,"Secure outsourcing of scientific computations," Advances in Computers,vol. 54, pp. 216–272, 2001.

[2] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications: a review and open problems," in Proc. of New Security Paradigms Workshop (NSPW), 2001, pp. 13–22.

[3] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," in Proc. of STOC, 2008, pp.113–122.Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, online at http://www.cloudsecurityalliance.org.

[4] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in Proc. Of CRYPTO'10, Aug. 2010.

[5] Sun Microsystems, Inc., "Building customer trust in cloud computing with transparent security," 2009, online at https://www.sun.com/offers/details/sun transparency.xml..

[6] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. of ICDCS'10, 2010.

[7] J. Li and M. J. Atallah, "Secure and private collaborative linear programming," in Proc.of CollaborateCom, Nov. 2006.

[8] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, online at http://www.cloudsecurityalliance.org.

[9] C. Gentry, "Computing arbitrary functions of encrypted data," Commun. ACM, vol. 53, no. 3, pp. 97–105, 2010.

[10] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc of STOC, 2009, pp. 169–178.

AUTHOR'S PROFILE

G. ASHOK KUMAR Pursuing M.Tech (CSE), QIS College of Engineering and Technology Vengamukkalapalem, Ongole, Prakasham Dist, Andhra Pradesh, India. His researches interests include cloud computing and linear Programming.

P. SRINIVASULU received his B. Tech from Acharya Nagarjuna University, Guntur, Andhra Pradesh in 1994 and completed post graduation from Jawaharlal Nehru Technological University, Hyderabad in 1998. He is received Ph.D from Acharya Nagarjuna University, Guntur and working as Professor in QIS College of Engineering and Technology, in the Department of Computer Science and Engineering, Vengamukkalapalem, Ongole, Prakasham Dist, Andhra Pradesh. His research interests include Data Mining and Data Warehousing, Computer Networks, Network security and Parallel Computing. He has more than 16 years of experience in teaching in many subjects, industry and in research. He is the member of Indian Society of Technical Education (ISTE) and also member of Computer Society of India (CSI). He has many publications in National and International conferences.