International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.3, Issue.3, May-June. 2013 pp-1396-1399 ISSN: 2249-6645

Secure Outsourcing Mechanism for Linear Programming in Cloud Computing

Shaik Kalesha, [1] G.Sridevi[2]
[1]M.Tech, Nimra College of Engineering & Technology, Vijayawada, A.P., India
[2]Assoc.Professor, Dept.of CSE, Nimra College of Engineering & Technology, Vijayawada, A.P., India

Abstract: *Cloud Computing is a subscription based service where you can obtain the networked storage space and computer resources. In cloud computing model, the customers plug into the cloud to access IT resources which are priced and provided on demand services. This cloud computing model composed of five essential characteristics, three service models and four deployment models. Users can store their data in the cloud and there is a lot of personal information and potentially secure data that people store on their computers, and this information is now being transferred to the cloud. Here we must ensure the security of user's data, which is in stored in the cloud. This paper presents the secure outsourcing mechanism for linear programming in the cloud computing environment. 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 the engineering optimization. It has been widely used in various engineering disciplines that analyze and optimize real world systems, such as - packet routing, flow control, power management of data centers, etc.*

Keywords: *Cloud, Linear programming, Security.*

I. INTRODUCTION

The end of this decade is marked by a paradigm shift of the industrial information technology towards the subscription based or pay-per-use service business model known as cloud computing [1]. This paradigm provides users with a long list of advantages, such as- provision computing capabilities; broad, heterogeneous network access; resource pooling and rapid elasticity with measured services [2]. Huge amounts of data being retrieved from the geographically distributed data sources, and non-localized data-handling requirements, creates such a change in technological as well as business model. One of the prominent services offered in the cloud computing is the cloud data storage, in which subscribers do not have to store their data on their own servers, where instead their data will be stored on the cloud service provider's servers.

In cloud computing, the subscribers have to pay the service providers for this storage service. This service does not only provides flexibility and the scalability for the data storage, it also provide customers with the benefit of paying only for the amount of data they need to store for a particular period of time, without any concerns for efficient storage mechanisms and maintainability issues with large amounts of data storage. In addition to these benefits, the customers can easily access their data from any geographical region where the Cloud Service Provider's network or Internet can be accessed. An example of the cloud computing is shown in Figure 1.

Along with these unprecedented advantages, the cloud data storage also redefines the security issues targeted on customer's outsourced data (data that is not stored/retrieved from the costumers own servers). Since the cloud service providers (SP) are separate market entities, data integrity and privacy are the most critical issues that need to be addressed in cloud computing. Even though the cloud service providers have standard regulations and has powerful infrastructure to ensure customer's data privacy and provide a better availability, the reports of privacy breach and service outage have been apparent in last few years [3] [4] [5].

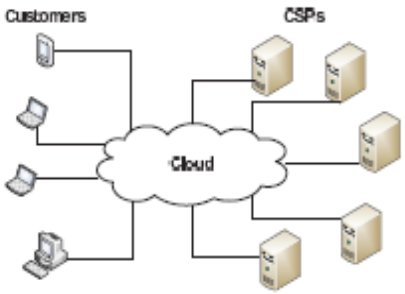

Figure 1: Cloud Computing Architecture Example

This paper presents the secure outsourcing mechanism for linear programming (LP) in the cloud computing environment. 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 the engineering optimization. It has been widely used in various engineering disciplines that analyze and optimize real world systems, such as - packet routing, flow control, power management of data centers, etc.

II. RELATED WORK

General secure computation outsourcing that fulfills all the aforementioned requirements, such as input/output privacy and correctness/soundness guarantee has been shown feasible in theory by Gennaro et al. [6]. However, it is currently not practical due to the huge computation complexity. Atallah et al. explore a list of work [7][8] for securely outsourcing specific applications. The customized solutions are expected to be very efficient than the general way of constructing the circuits. In [7], they give the first investigation of secure outsourcing of numerical and scientific computation. Later on in [8] and [9], Atallah et al. give two protocol designs for both secure sequence comparison outsourcing and the secure algebraic computation outsourcing. However, both protocols use heavy cryptographic primitive such as homomorphic encryptions [10] and/or oblivious transfer [11] and do not scale well for large problem set.

Hohenberger et al. [12] provide protocols for secure outsourcing of modular exponentiation, which is considered as prohibitively expensive in most public-key cryptography operations. Recently, Atallah [13] et al. give a provably secure protocol for secure outsourcing matrix multiplications based on secret sharing [14]. Another large existing list of work that relates to ours is Secure Multi-party Computation (SMC), first introduced by Yao [15] and later extended by Goldreich et al. [16] and many others. Very recently, Wang et al. [17] give the first study of secure outsourcing of linear programming in cloud computing. Their solution is based on the problem transformation, and has the advantage of bringing customer savings without introducing substantial overhead on cloud. However, those techniques involve cubic time computational burden matrix-matrix operations, which the weak customer in our case is not necessarily able to handle for large-scale problems.

III. PROPOSED WORK

Our proposed mechanism consists of three phases:

A. Problem Transformation

In this phase, the cloud customer would initialize a randomized key generation algorithm and prepare the LE problem into some encrypted form ϕ_K via key K. Transformation and encryption operations will be needed when necessary.

The customer who has coefficient vector "b" and seeks solution "x" satisfying Ax = b cannot directly starts the ProbSolve with cloud, since such interaction may expose the private information on final result x. Thus, we still need a transformation technique to allow the customer to properly hide such information first. The customer picks a random vector r $\in R^n$ as his secret keying material, the new LE problem is written as:

$$Ay = b' \quad \text{---------- (1)}$$

Where y = x + r and b' = b + Ar. Equation (1) can be rewritten as follows:

$$y^{K-1} = T.y^{(k)} + C' \quad \text{---------- (2)}$$

Where T= $D^{-1}.R$, $c' = D^{-1}.b'$, and A= D + R.

The whole procedure of "ProbTransform" is summarized in the following Algorithm 1.

Algorithm1:
Data: original problem Φ =(A,b)
Result: transformed problem as shown in Equation (2)

Step 1: Pick random r $\in R^n$
Step 2: Compute b^1 =b+Ar, and $c^1 = D^{-1}.b^1$
Step 3: Replace tuple (x,c) in Equation (2) with (y=x+r, c^1)
Return transformed problem as Equation (2).

B. Problem Solving

In this phase, the cloud customer would use the encrypted form ϕ_K of LE to start the computation outsourcing process. In case of using the iterative methods, the protocol ends when the solution within the required accuracy is found.

After the problem transformation step, now we are ready to describe the phase of "ProbSolve". The purpose of the protocol is to let the customer securely harness the cloud for the most expensive computation, i.e., the matrix-vector multiplication T. $y^{(K)}$ in Eq. (2) for each algorithm iteration, k = 1, 2,……L. assume without loss of generality that our main protocol of solving LE works over integers. All arithmetic is modular with respect to the modulus "N" of the homomorphic encryption, and the modulus is large enough to contain the answer. For the first iteration, the customer starts

the initial guess on the vector $y^{(0)} = (y1^{(0)}, y2^{(0)}, \ldots\ldots yn^{(0)})^T$, and then sends it to the cloud. The cloud server, in possession of the encrypted matrix Enc (T), computes the value Enc (T. $y^{(0)}$) by using the homomorphic property of the encryption:

$$\mathrm{Enc}(\mathbf{T} \cdot \mathbf{y}^{(0)})[i] = \mathrm{Enc}(\sum_{j=1}^{n} \mathbf{T}[i,j] \cdot y_j^{(0)})$$

$$= \prod_{j=1}^{n} \mathrm{Enc}(\mathbf{T}[i,j])^{y_j^{(0)}} \quad \text{------- (3)}$$

After receiving Enc (T. $y^{(0)}$) the customer decrypts and gets value T. $y^{(0)}$ Using his private key. He then updates the next approximation $y^{(1)}$ = T. $y^{(0)}$ + $c^{'}$ via Equ (2). The protocol execution continues until the result converges, as shown in the following Algorithm 2.

Algorithm 2:

Data: Transformed problem with input c^1 and Enc (T)
Result: Solution x to the original problem Φ = (A, b)
% L: Maximum number of iterations to be performed;
% $\in$: Measurement of convergence point;

Step 1: Customer picks $y^{(0)} \in (Z_N)^N$
For (k<-0 to L) do

Step 2: Customer sends $y^{(k)}$ to cloud

Step 3: Cloud computes Enc (T $y^{(k)}$) via Equation (3)

Step 4: Customer decrypts T $y^{(k)}$ via his private key
If $\| y^{(k)} - y^{(k+1)} \| <= \in$ then

Step 5: break with convergence point $y^{(k+1)}$

Step 6: return x= $y^{(k+1)}$ - r

C. Result Verification

In this phase, the cloud customer would verify the encrypted result produced from the cloud server, using the randomized secret key K. A correct output "x" to the problem is produced by decrypting the encrypted output. When the validation fails, the customer output $\perp$, indicating the cloud server was cheating.

IV. CONCLUSION

Cloud Computing provides convenient on demand network access to a shared pool of configurable computing resources that can be rapidly deployed with the great efficiency and minimal management overhead. Focusing on the engineering and scientific computing problems, this proposed work investigates secure outsourcing for widely applicable large-scale systems of linear equations (LE), which are among the most popular algorithmic and computational tools in various engineering disciplines that analyze and optimize real-world systems. Our proposed work has three phases- problem transformation, problem solving and result verification. In problem transformation, the cloud customer would initialize a randomized key generation algorithm and prepare the LE problem into some encrypted form ϕ_K via key K. Transformation and encryption operations will be needed when necessary. In problem solving, the cloud customer would use the encrypted form ϕ_K of LE to start the computation outsourcing process. In case of using the iterative methods, the protocol ends when the solution within the required accuracy is found. In result verification, the cloud customer would verify the encrypted result produced from the cloud server, using the randomized secret key K.

REFERENCES

[1] M. Armbrust, A. Fox, R. Grifth, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, University of California at Berkeley, February 2009.
[2] P. Mell, T. Grance, "Draft NIST working definition of cloud computing", Referenced on June. 3rd, 2009, online at http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, 2009.
[3] Amazon.com, "Amazon s3 availablity event: July 20, 2008", online at http://status.aws.amazon.com/s3-20080720.html, 2008.
[4] M. Arrington, "Gmail Disaster: Reports of mass email deletions", Online at http://www.techcrunch.com/2006/12/28/gmail-disasterreports-ofmass- email-deletions/, December 2006.
[5] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors", Online at http://www.techcrunch.com/2008/-7/10/mediamaxthelinkup-closes-itsdorrs/, July 2008
[6] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in Proc. Of CRYPTO'10, Aug. 2010.
[7] 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.
[8] M. J. Atallah and J. Li, "Secure outsourcing of sequence comparisons," Int. J. Inf. Sec., vol. 4, no. 4, pp. 277–287, 2005
[9] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in Proc. of 6th Conf. on Privacy, Security, and Trust (PST), 2008, pp. 240–245.
[10] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Proc. of EUROCRYPT'99, 1999, pp. 223–238.
[11] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," Commun ACM, vol. 28, no. 6, pp. 637–647, 1985.
[12] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations" in Proc. of TCC, 2005 pp. 264–282.
[13] M. Atallah and K. Frikken, "Securely outsourcing linear algebra computations," in Proc. of ASIACCS, 2010, pp. 48–59
[14] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979
[15] A. C.-C. Yao, "Protocols for secure computations (extended abstract)," in Proc. of FOCS'82, 1982, pp 160–164
[16] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority," in Proc. of STOC'87, 1987, pp 218–229
[17] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in Proc. of IEEE INFOCOM, 2011