ISSN: 2320-1363

Practical Outsourcing of LP in Cloud Computing For Information Secure

N.Mahita*1, P Ramesh Babu*2

M.Tech (CSE) Student Department of CSE, Priyadarshini Institute of Technology & Science, Chintalapudi, Guntur(Dist), Ap, India.

Assistant Professor, Department of CSE in Priyadarshini Institute of Technology & Science, Chintalapudi, Guntur(Dist), Ap, India

Abstract

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. But often the information housed on the cloud is often seen as valuable to individuals with malicious intent. 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 article 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. In this the 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.

Keywords: cloud computing, confidential information, homomorphic encryption, result verification, LP computation.

1. Introduction

Cloud computing is broken down into three segments: "application" "storage" and "connectivity." Each segment serves a different purpose and offers different products for businesses and individuals around the world. You may know that there are airports and other planes out there, somewhere in those clouds. You have to trust that the sky is big, and that there is room out there for everyone, and that people you can trust are in control, and that those people can see perfectly in and through and around all the clouds. In most every diagram, a cloud icon represents I.T. resources that are not inside the four walls of the company's enterprise. The cloud icon is meant to simply say, "Look, there are parts of our Information Technology system that we don't really control.

Cloud computing changes the way we think about technology. Cloud computing allows

ISSN: 2320-1363

IT organization to increase agility and often lower costs. This allows them to anticipate and respond to business needs more effectively and thereby create competitive advantage. The main advantage of cloud architectures are : reduced cost, cloud computing can reduce both capital expense and operating expense costs because resources are only acquired when needed and are only paid for when used. Refined usage of personal: using cloud computing frees valuable personal allowing them to focus on delivering value rather than maintaining hardware and software. Robust stability: cloud computing allows for immediate scaling either up or down, at any time without long-term commitment. By outsourcing organizations can concentrate on their own task and operate other business applications via the Internet rather than incurring substantial hardware, software and personal costs involved in maintaining application in house. A well proven scheme of protecting the sensitive file is to pre-compute the cryptographic hash of F using $h_k(F)$ and to store this hash as well as the secret key K. the verifier is responsible to do the verification.

2. Related Work

In this work the verifier does the computation steps before only archiving the file F in the cloud storage. It is also worth to use only one cryptography key irrespective of the size of the file or the number of files whose retrievability needs to be verified. The advantage of this approach is that the archive needs to access only a small portion of the file F unlike the previous key-hash scheme where the full file had to be archived for each and every verification.

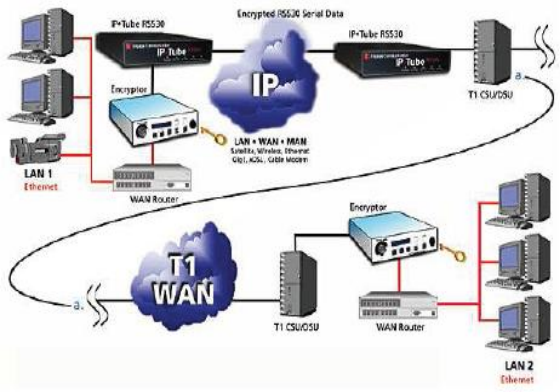

Fig.2.1. An example usage area of cloud computing.

In this scheme special file blocks called sentinels are hidden among other blocks in the data file F. this kind of cryptographic computation prevent cloud from invoking any kind of operation on the plain text file or data as computations on encrypted data is a tough one.

Linear programming is an algorithm and computational tool which capture the first order effect of various system parameters that should be optimized, and is essential to engineering optimization. As LP computations need enough computational power and involve confidential data, in this article 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

ISSN: 2320-1363

by the customer. 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 form 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.

3. Proposed system architecture

The proposed system model of cloud computing environment consists of total seven main components, namely customer linear programmed data, method design, key generation, problem encryption, proof generation and key description.

Each of the following components are doing their individual work to make the system successful as follows:

Customer: The computation tool is essential to engineering optimization. Here the customer uses this computational tool for capturing first order effect of various system parameters which must be optimized.

Linear programming data: After verifying the result customer uses the confidential translation to map back the desired solution for his original LP problem.

Key generation: 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.

Encryption: This algorithm encrypts the input tuple γ into γ_k with the secret key K. according to problem transformation the encrypted input γ_k has the same form as γ and thus defines the problem to be solved in the cloud.

Prof Generation: This algorithm augments a generic solver that solves the problem $\square_k$ to produce both the output μ and a proof ∂. The output later decrypt to €, and ∂ is used later by the customer to verify the correctness of μ or €.

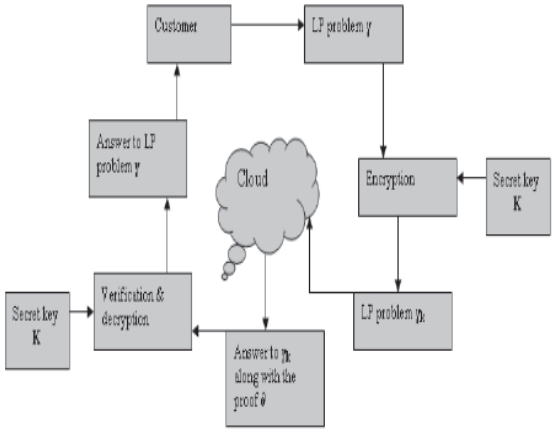

Fig.3.1 proposed system architecture.

Key description: It must produce an output that can be decrypted and verified successfully by the customer.

4. 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 article is completely

ISSN: 2320-1363

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 $(1^K) \rightarrow \{K\}$: 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 $(K, \gamma) \rightarrow \{\gamma_k\}$: This algorithm encrypts the input tuple γ into γ_k with the secret key K. according to problem transformation, the encrypted input γ_k has the same form as γ and thus defines the problem to be solved in the cloud.

ProffGen $(\gamma_k) \rightarrow \{(\mu, \partial)\}$: this algorithm augments a generic solver that solves the problem γ_k to produce both the output μ and a proof ∂. The output μ later decrypts to €, and ∂ is used later by the customer to verify the correctness of μ or €.

ResultDec $(K, \gamma, \mu, \partial) \rightarrow \{€,\#\}$: This algorithm may choose to verify either μ or €. Via the proof ∂. In any case, a correct output € is produced by decrypting μ 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 $\mathbf{K}= (\mathbf{Q}, \boldsymbol{\lambda}, \boldsymbol{\theta})$ and encrypt the input tuple γ into $\gamma_k = (\mathbf{A'}, \mathbf{B'}, \mathbf{b'}, \boldsymbol{\theta}_c)$, 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. Essentially it shows that although one can change the constraints to a completely different from it is not necessary the feasible region defined by the constraint 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.

5. Result and verification

Initially we assume that the server performs the computation honestly. Although this kind of semi honest model is not strong enough to capture the adversary behaviors in the real world. In most of the situations, when the cloud side computation requires a huge amount of computing resources, there exists a strong financial commitment which insists the cloud server to become "lazy". They may not be willing to commit service-level-agreed computing resources for saving cost, or even sometimes be malicious just only to sabotage any following up computation at the customers. Since the cloud server promises to solve the LP problem $\boldsymbol{\gamma_k} = (\mathbf{A'}, \mathbf{B'}, \mathbf{b'}, \mathbf{c'})$, we propose to solve the result verification problem by designing a method to verify the correctness of the solution $\boldsymbol{\mu}$ of $\boldsymbol{\gamma_k}$. The soundness condition would be a corollary thereafter when we present the whole mechanism in

ISSN: 2320-1363

the next section. Here the workload required for customers on the result verification is substantially cheaper than solving the LP problem on their own, which ensures the great computation savings for secure LP outsourcing. The LP problem does not necessarily have an optimal solution. The LP problem does not necessarily have an optimal solution. The two possible cases are as follows:

- **Usual scenario**:

There is an optimal solution with finite objective value. We first assume that the cloud server returns an optimal solution μ. In order to verify μ without actually solving the LP problems, we design our method by seeking a set of necessary and sufficient conditions that the optimal solution must satisfy. The derivative of these conditions from the well studied duality theory of the LP problem. For the primal LP problem γ_k its dual problem is defined as:

Maximize $b'^{T}s$ subject to $A'^{T}s+B'^{T}t = c', t > 0$. (1)

Where s and t are the m x 1 and n x 1 vectors of dual decision variable respectively.

The strong duality of the LP problems states that if a primal feasible solution **μ** and a dual feasible solution **(s, t)** lead to the same primal and dual objective value, then both **μ** and **(s, t)** are the optimal solutions of the primal and the dual problems respectively. We should ask the cloud server to provide the dual optimal solution as part of the proof *∂*. Then, the correctness of **μ** can be verified based on the following conditions:

$C'^{T} y = b'^{T} s$, $A'^{Y} = b'$, $B'^{Y} \geq 0$, $A'^{T} s + B'^{T} t = c'$, $t \geq 0$ (2)

Here $C'^{T}\mu = b'^{T} s$ tests the equivalence of primal and dual objective value for strong duality. All the remaining conditions ensure that both μ and (s, t) are feasible solutions of the primal and dual problems, respectively.

- **Unbounded scenario**:

The result verification method not only needs to verify a solution if the cloud server returns one, but also needs to verify the cases when the cloud server claims that the LP problem is unbounded. For this kind of cases we will first present the proof *∂* that the cloud server should provide and the verification method when the cloud server returns an optimal solution, and then present the proofs and the methods for the other two cases, each of which is built upon the previous one. Finally, we assume that the cloud server claims $\boldsymbol{\gamma_k}$ to be unbounded. The duality theory implies that this case is equivalent to that $\boldsymbol{\gamma_k}$ is feasible and the dual problem of $\boldsymbol{\gamma_k}$, i.e. Eq. (1) is infeasible. Therefore, the cloud server should provide a proof showing that those two conditions hold. For all the above cases, the cloud server is required to provide corrections proof by proving a normal LP has an optimal solution.

6. Conclusion

We formalize the problem of security outsourcing LP computations in cloud computing and provide such a practical mechanism design which fulfills ip/op privacy, cheating resilience, and efficiency. In this paper we show that for any problem γ and its encrypted version γ_k, solution μ

ISSN: 2320-1363

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. By explicitly decomposing LP computation outsourcing into public LP solvers and private data our mechanism design is able to explore appropriate security or efficiency tradeoffs via higher level LP computation than the general circuit representation. We develop problem transformation techniques that enable customer to secretly transform the original LP into some arbitrary one while protecting sensitive input/output information. In near future a goal is set to work around some interesting concepts such as to devise robust algorithms to achieve num erical stability to explore the sparsity structure of problem for further efficiency improvement, to establish formal security frame work and also to extend our result to nonlinear programming computation outsourcing in cloud.

References

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

[2] Sanchari Saha, " preserving data accuracy in the cloud infrastructure", in proceedings of national conference- NACC-2012, May 10-11, 2012, Bangalore, India.

[3] Cong Wang, Kui Ren, and Jia Wang, "Secure and practical outsourcing of linear programming in cloud computing", IEEE transactions on cloud computing April- 10-15, 2011.

[4] D. Luenberger and Y. Ye, Linear and Nonlinear Programming, 3rd ed. Springer, 2008.

[5] 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.

[6] S. Hohenberger and A. Lysyanskaya, "How to securely outsourcevcryptographic computations," in Proc. of TCC, 2005, pp. 264–282.

[7] M. J. Atallah and J. Li, "Secure outsourcing of sequence comparisons," Int. J. Inf. Sec., vol. 4, no. 4, pp. 277–287, 2005.

[8] 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.

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

[10] M. Atallah and K. Frikken, "Securely outsourcing linear algebra computations," in Proc. of ASIACCS, 2010, pp. 48–59.

[11] A. C.-C. Yao, "Protocols for secure computations (extended abstract)," in Proc. of FOCS'82, 1982, pp. 160–164.