Linear and Non Linear Programming in Cloud Computing

[1]P. Aslesha, [2]P. Radha Krishna, [3]S. Rajitha
[1,2,3]Dept. of CSE, NOVA College of Engineering & Technology, Vijayawada, AP, India

Abstract
In this paper, for the first time, we formalize the problem of securely outsourcing LP computations in cloud computing, and provide such a practical mechanism design which fulfills input/output privacy, cheating resilience, and efficiency. By explicitly decomposing LP computation outsourcing into public LP solvers and private data, our mechanism design is able to explore appropriate security/efficiency tradeoffs via higher level LP computation than the general circuit representation. We develop problem transformation techniques that enable customers to secretly transform the original LP into some arbitrary one while protecting sensitive input/output information. 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 demonstrates the immediate practicality of the proposed mechanism. We plan to investigate some interesting future work as follows:
1. Devise robust algorithms to achieve numerical stability
2. Explore the sparsity structure of problem for further efficiency improvement
3. Establish formal security framework
4. Extend our result to non-linear programming computation outsourcing in cloud.

Keywords
Integrity, Conditional Functional Dependency, Functional Dependency, Free Item Set, Closed Item Set

I. Introduction

Increasingly, scientific and commercial applications are leveraging the power of distributed computing and storage resources [4, 15]. These resources are available either as part of general purpose computing infrastructure such as Clusters and Grids, or through commercially hosted services such as Clouds [1]. Buyya et al. [3], have defined Clouds to be a type of parallel and distributed system consisting of inter-connected and virtualized computers. These computers can be dynamically provisioned as per users' requirements. Thus, to achieve better performance and scalability, applications could be managed using commercial services provided by Clouds, such as Amazon AWS, Google AppEngine, and Microsoft Azure. However, the cost of computing, storage and communication over these resources could be overwhelming for compute intensive and data-intensive applications. Data mining is one example application domain that comprises of data-intensive applications often with large distributed data and compute-intensive tasks to manage data and knowledge distribution. Examples of data mining applications are: checking bank account lists with lists of suspected criminals (Watch List Compliance), checking for duplication of customer data in financial marketing, using catalogue data in astrophysical image analysis or detecting the spread of Internet worms using intrusion detection systems. The data to be mined may be widely distributed depending on the nature of the application. As the size of these datasets increases over time, the analysis of distributed data-sets on computing resources for multiple users (repeated executions) has the following challenges:

1. Large number of data-sets and mining tasks make the application complex! requires a well-designed application workflow
2. Large size and number of distributed data-sets make the application data-intensive ! requires minimization of communication and storage costs
3. Cost of computing (classification/knowledge discovery) and transferring of data increases as the number of iterations/data-sets increase ! requires minimization of repeated data mining costs In this paper, we address the challenges listed above for data-intensive workflows by making the following three contributions:

- We model the cost of execution of a workflow on Cloud resources using a Non-Linear Programming (NLP) model. The NLP-model retrieves data partially from multiple data sources based on the cost of transferring data from those sources to a compute resource, so that the total cost of data-transfer and computation cost on that compute resource is minimized.
- We take Intrusion detection as a data mining application for a case study. This application has all the features as listed in the previous paragraph when executing commercially [15]. We design the application as a workflow that simplifies the basic steps of data mining into blocks.3. We then apply the NLP-model on the intrusion detection application to minimize repeated execution costs when using commercial compute and storage resources. We compare the cost when using Amazon Cloud Front with simple round-robin scheduling algorithm.

II. Existing System

In this section we discuss about Linear programming We consider a computation outsourcing architecture involving two different entities, as illustrated the cloud customer, who has large amount of computationally expensive LP problems to be outsourced to the cloud; the Cloud Server (CS), which has significant computation resources and provides utility computing services, such as hosting the public LP solvers in a pay-per-use manner. The customer has a large-scale linear programming problem _ (to be formally defined later) to be solved. However, due to the lack of computing resources, like processing power, memory, and storage etc., he cannot carry out such expensive computation locally. Thus, the customer resorts to CS for solving the LP computation and leverages its computation capacity in a pay-per-use manner. Instead of directly sending original problem _, the customer first uses a secret K to map _ into some encrypted version _K and outsources problem _K to CS. CS then uses its public LP solver to get the answer of _K and provides a correctness proof □, but it is supposed to learn nothing or little of the sensitive information contained in the original problem description _. After receiving the solution of encrypted problem _K, the customer should be able to first verify the answer via the appended proof □. If it's correct, he then uses the secret K to map the output into the desired answer for the original problem _. The security threats faced by the computation model primarily come from the malicious behavior of CS. We assume that the CS may behave beyond "honest-but-curious", i.e. the semihonest model that was assumed by many previous researches

(e.g., [14-15]), either because it intends to do so or because it is compromised. The CS may be persistently interested in analyzing the encrypted input sent by the customer and the encrypted output produced by the computation to learn the sensitive information as in the semi-honest model. In addition, CS can also behave nfaithfully or intentionally sabotage the computation, e.g. to lie about the result to save the computing resources, while hoping not to be caught at the same time. Finally note that we assume the communication channels between each cloud server and the customer is authenticated and reliable, which can be achieved in practice with little overhead. These authentication handshakes are omitted in the following presentation. B. Design Goals To enable secure and practical outsourcing of LP under the aforementioned model, our mechanism design should achieve the following security and performance guarantees.

A. Correctness

Any cloud server that faithfully follows the mechanism must produce an output that can be decrypted and verified successfully by the customer.

B. Soundness

No cloud server can generate an incorrect output that can be decrypted and verified successfully by the customer with non-negligible probability

C. Input/output Privacy

No sensitive information from the customer's private data can be derived by the cloud server during performing the LP computation.

D. 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. C. Background on Linear Programming An optimization problem is usually formulated as a mathematical programming problem that seeks the values for a set of decision variables to minimize (or maximize) an objective function representing the cost subject to a set of constraints For linear programming, the objective function is an affine function of the decision variables, and the constraints are a system of linear equations and inequalities. Since a constraint in the form of a linear inequality can be expressed as a linear equation by introducing a non-negative slack variable, and a free decision variable can be expressed as the difference of two non-negative auxiliary variables, any linear programming problem can be expressed in the following standard form, minimize cT x subject to Ax = b, x ≥ 0. (1)

Here x is an n×1 vector of decision variables, A is an m×n matrix, and both c and b are n×1 vectors. It can be assumed further that m ≤ n and that A has full row rank; otherwise, extras rows can always be eliminated from A.

In this paper, we study a more general form as follows, minimize cT x subject to Ax = b,Bx ≥ 0. (2) In Eq. (2), we replace the non-negative requirements in Eq. (1) by requiring that each component of Bx to be non-negative, where B is an n × n non-singular matrix, i.e. Eq. (2) degenerates to Eq. (1) when B is the identity matrix. Thus, the LP problem can be defined via the tuple _ = (A,B, b, c) as input, and the solution x as output.

III. Proposed System

In this section we discuss about the Nonlinear programming. Here, we try to get the minimum cost by formulating a non-linear program for the cost-optimization problem, as depicted. The formulation uses two variables y, d and pre-computed values txcost, ecost, txtime, etime as listed below:

• y characterizes where each task is processed. Yk j = 1 iff task Tk is processed on processor P_j.

• d characterizes the amount of data to be transferred to a site. e.g. dk i,j = 50.2 denotes 50.2 units of data are to be transferred from Si $ Pj for task Tk.

• txcost characterizes the cost of data transfer for a link per data unit. e.g. txcosti,j = 10 denotes the cost of data transfer from Si $ Pj . It is added to the overall cost iff dk i,j > 0 & yk j = 1.

• ecost characterizes the cost of computation (usage time) of a processor. e.g. ecostj = 1 denotes the cost of using a processor Pj. It is added to the overall cost iff yk j = 1.

• txtime characterizes the average time for transferring unit data between two sites. e.g. txtimei,j = 50 denotes the time for transferring unit data from Si $ Pj . It is added to the Execution Time (ET) for every task iff dk i,j > 0 & yk j = 1.

• etime characterizes the computation time of a task averaged over a set of known and dedicated resources. e.g. etimekj = 20 denotes the time for executing a task Tk on a processor Pj . It is added to ET iff yk j = 1. The constraints can be described as follows:

• (a) & (h) ensure that each task k " T is computed only once at processor j " P when the variable yk j > 0. For partial values of yk j , we round up/down to the nearest integer (0 or 1). Tasks are not partitioned or migrated.

• (b) & (c) ensure that partial data transferred and total data required by a task cannot be negative.

• (d), (e), (f) and (g) ensure that cost and time values are all positive.

• (i), (a) & (b) ensure that partial-data are transferred only to the resource where a task is executed. For all such transfers, the sum of data transferred should equal to the data required by the task, which is tdatak.

• (j) ensures that the total data transfer for all the tasks are bounded by the sum of data required by each task. This is important for the solvers to relate (h), (i) & (j).

• (i) & (j) combined ensure that whenever partial-data dk i,j is transferred to a compute host Pj , then a compute host must have been selected at j (yk j = 1), and that total data transfer never exceeds the bound tdatak for each task and in total. To get an absolute minimum cost, we map the tasks in the workflow onto resources based only on cost optimization (not time). This eliminates the time dependencies between tasks. However, the task to compute-resource mappings and data-source to compute-resource mappings minimizes the cost of execution but not the make span. The execution time of a task (ET k) is calculated based on the c(ost-minimized mappings given by the solver. The total: k$T (ETk+waiting time) is the make span of the workflow with the minimum cost, where the waiting time denotes the minimum time a task has to wait before its parents finish execution.

$$
\begin{aligned}
&\textit{Minimize total Cost } (C)\\
&C = \sum_{i\in S, j\in P, k\in T} d_{i,j}^{k} * txcost_{i,j} * y_j^k + ecost_j * etime_j^k * y_j^k\\
&\textit{Subject to :}\\
&\quad (a)\ \forall k \in T, j \in P \qquad y_j^k \geq 0\\
&\quad (b)\ \forall i \in S, j \in P, k \in T \quad d_{i,j}^{k} \geq 0\\
&\quad (c)\ \forall k \in T \qquad tdata^k \geq 0\\
&\quad (d)\ \forall i \in S, j \in P \qquad txcost_{i,j} \geq 0\\
&\quad (e)\ \forall i \in S, j \in P \qquad txtime_{i,j} \geq 0\\
&\quad (f)\ \forall k \in T, j \in P \qquad ecost_j \geq 0\\
&\quad (g)\ \forall k \in T, j \in P \qquad etime_j^k \geq 0\\
&\quad (h)\ \sum_{j\in P} y_j^k = 1\\
&\quad (i)\ \sum_{i\in S, j\in P} y_j^k * d_{i,j}^{k} = tdata^k\\
&\quad (j)\ \sum_{i\in S, j\in P, k\in T} y_j^k * d_{i,j}^{k} = \sum_{k\in T} tdata^k\\
&\textit{Execution time of task } k\ (ET^k)\\
&ET^k = \sum_{i\in S, j\in P, k\in T} (d_{i,j}^{k} * txtime_{i,j} * y_j^k) + etime_j^k * y_j^k
\end{aligned}
$$

Fig. 1:

IV. Results

In order to implements these four algorithms we calculate the performance of algorithms on the basis on time complexity and space complexity

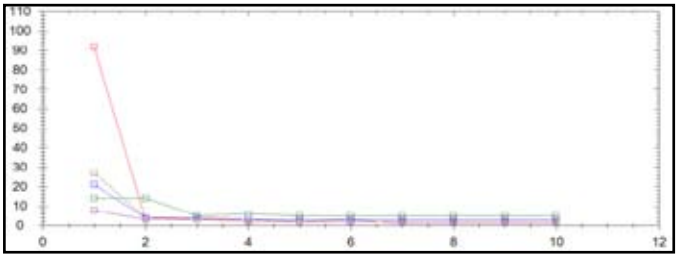


Fig. 2:

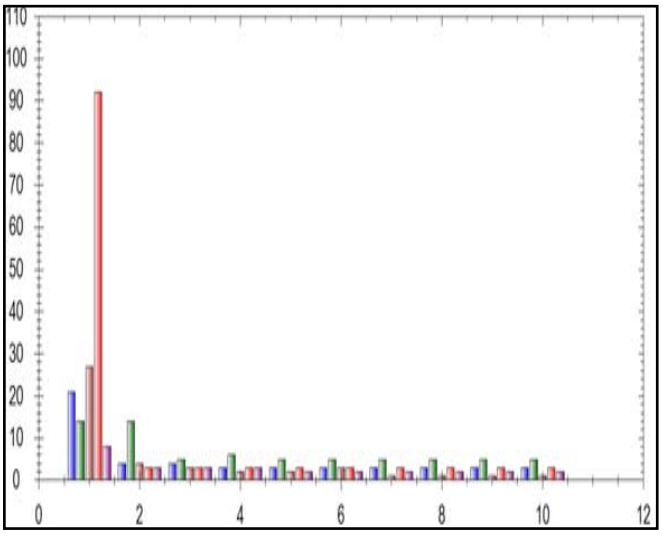


Fig. 3:

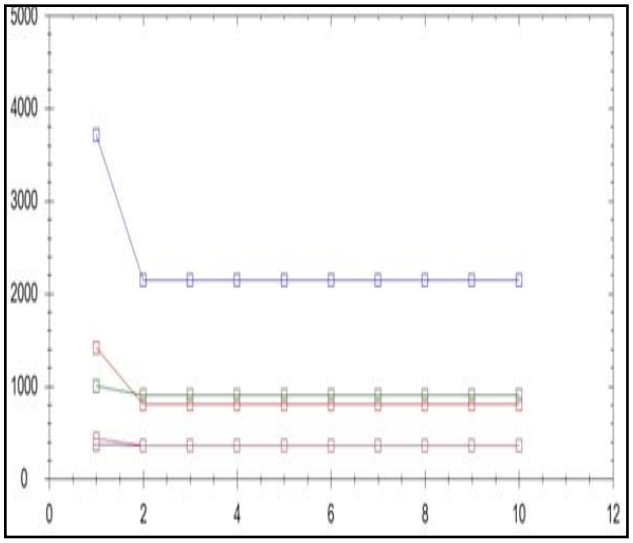


Fig. 4:

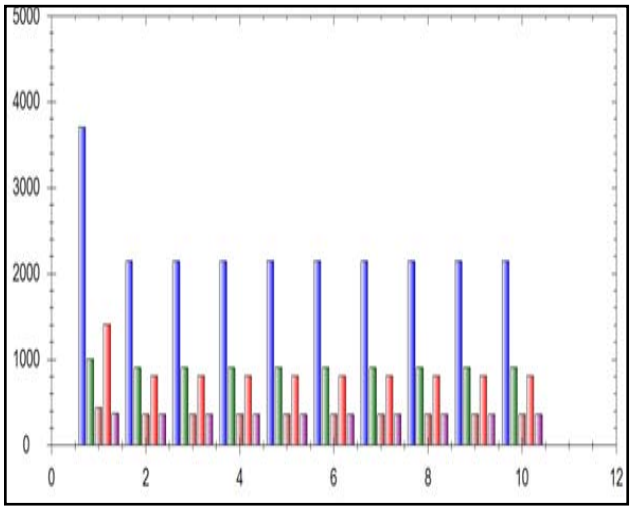


Fig. 5:

V. Conclusions and Future Work

Cloud Computing has great potential of providing robust computational power to the society at reduced cost. It enables customers with limited computational resources to outsource their large computation workloads to the cloud, and economically enjoy the massive computational power, bandwidth, storage, and even appropriate software that can be shared in a pay-per-use manner. Despite the tremendous benefits, 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. Treating the cloud as an intrinsically insecure computing platform from the viewpoint of the cloud customers, we must design mechanisms that not only protect sensitive information by enabling computations with encrypted data, but also protect customers from malicious behaviors by enabling the validation of the computation result. Such a mechanism of general secure computation outsourcing was recently shown to be feasible in theory, but to design mechanisms that are practically efficient remains a very challenging problem. Focusing on engineering computing and optimization tasks, this paper investigates secure outsourcing of widely applicable Linear Programming (LP) computations. In order to achieve practical efficiency, our mechanism design explicitly decomposes the LP computation outsourcing into public LP solvers running on the cloud and private LP parameters owned by the customer. The

resulting flexibility allows us to explore appropriate security/ efficiency tradeoff via higher-level abstraction of LP computations than the general circuit representation. In particular, by formulating private data owned by the customer for LP problem as a set of matrices and vectors, we are able to develop a set of efficient privacy-preserving problem transformation techniques, which allow customers to transform original LP problem into some arbitrary one while protecting sensitive input/output information. To validate the computation result, we further explore the fundamental duality theorem of LP computation and derive the necessary and sufficient conditions that correct result must satisfy. Such result verification mechanism is extremely efficient and incurs close-to-zero additional cost on both cloud server and customers. Extensive security analysis and experiment results show the immediate practicability of our mechanism design.

References

[1] P. Mell, T. Grance (2010),"Draft nist working definition of cloud computing", [Online] Available: http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html

[2] Cloud Security Alliance (2009),"Security guidance for critical areas of focus in cloud computing", [Online] Available: http://www.cloudsecurityalliance.org.

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

[4] Sun Microsystems, Inc.(2009),"Building customer trust in cloud computing with transparent security", [Online] Available: https://www.sun.com/offers/details/sun transparency.xml.

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

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

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

[8] D. Benjamin, 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, B. Parno,"Non-interactive verifiable computing:Outsourcing computation to untrusted workers", in Proc. of CRYPTO'10, Aug. 2010.

[10] M. Atallah 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.

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

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

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

[15] S. Yu, C. Wang, K. Ren, W. Lou,"Achieving secure, scalable, and fine-grained access control in cloud computing", in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March 2010.

[16] S. Boyd, L. Vandenberghe,"Convex Optimization", Cambridge University Press, 2004.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, "Introduction to Algorithms", 2nd ed. MIT press, 2008.

[18] V. Strassen,"Gaussian elimination is not optimal", Numer. Math., Vol. 13, pp. 354–356, 1969.

[19] D. Coppersmith and S. Winograd,"Matrix multiplication via arithmetic progressions", in Proc. of STOC'87, 1987, pp. 1–6.

[20] MOSEK ApS(2010),"The MOSEK Optimization Software", [Online] Available: http://www.mosek.com/

[21] P. Paillier,"Public-key cryptosystems based on composite degree residuosity classes", in Proc. of EUROCRYPT'99, 1999, pp. 223–238.

[22] S. Even, O. Goldreich, A. Lempel,"A randomized protocol for signing contracts", Commun. ACM, Vol. 28, No. 6, pp. 637–647, 1985.

[23] A. Shamir,"How to share a secret", Commun. ACM, Vol. 22, No. 11, pp. 612–613, 1979.

[24] O. Goldreich, S. Micali, 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.

[25] W. Du, 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.

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

[27] S. Goldwasser, Y. T. Kalai, G. N. Rothblum,"Delegating computation: interactive proofs for muggles", in Proc. of STOC, 2008, pp. 113–122.



P.Aslesha has completed B.Tech from NOVA College of Engineering Technology for Women in 2010 and Pursuing M.Tech in NOVA College of Engineering Technology for Women. Her research work includes concepts of cloud computing used for Linear and Non Linear Programming in Cloud Computing.



Mr. P.Radha Krishna did his Master in Computer Application at Lakkireddy Bali eddy college of Engineering. He started his teaching career at Anurag Engineering College, Kodad. Where He had guided be so many batches in their projects. He guided students in doing IBM projects at Anurag. He played an active role in the NBA Accreditation process at Anurag Engineering College in to 2007, He had acted as In charge HOD of MCA at Anurag. Mr. P Radha Krishna is a Members of ISTE. He had participated many workshops and paper presentations, which include sunmicrosystem and open systems (LAMP) workshops. Mr. Radha Krishna is the secretary of Alumni Association of LBRCE. He is now working as the Head of the Department of CSE of Nova college of Engineering & Technology, Jupudi, Ibrahimpatnm.



S.Rajitha has completed B.Tech from NOVA College of Engineering Technology for Women in 2010 and Pursuing M.Tech in NOVA College of Engineering Technology for Women. Her research work includes concepts of Data Mining and Data Warehousing used for Collective Behavior of social networking information by using Edge Clustering.