Securely Outsourcing Large Scale Systems of Linear Equations Using Homomorphic Encryption

Adilakshmy.T
Computer Science And Engineering
SCET,SGI
Villupuram,India
diviyabarati@gmail.com

Sharmila.H
Computer Science And Engineering
SCET,SGI
Villupuram,India
sharmisekar1992@gmail.com

Abstract— Harnessing the cloud for securely Outsourcing linear equations propose a very efficient cheating detection mechanism. In this paper, we formulate the problem in the computation outsourcing model for securely solving large-scale systems of LE via iterative methods, and provide the secure mechanism design which fulfills input/output privacy, cheating resilience, and efficiency. Our mechanism brings computational savings as it only incurs local computation burden for the customer within each algorithm iteration and demands no unrealistic IO cost, while solving large scale LE locally usually in terms of both time and memory requirements. We explore the algebraic property of matrix-vector multiplication to design a batch result verification mechanism, which allows customers to verify all answers computed by cloud from previous iterations in one batch, and further ensures both the efficiency advantage and the robustness of the design. 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.

***Index Terms*— Computation outsourcing, Iterative method, Computational savings, Homomorphic encryption, Cheating detection.**

I. INTRODUCTION

In cloud computing, customers can enjoy the literally unlimited computing resources in the cloud through the convenient yet flexible pay-per-use manners . Despite the tremendous benefits, the fact that customers and cloud are not necessarily in the same trusted domain brings many security concerns and challenges toward this promising computation outsourcing model. First, customer's data that are processed and generated during the computation in cloud are often sensitive in nature, such as business financial records, proprietary research data, and personally identifiable health information, etc. While applying ordinary encryption techniques to these sensitive information before outsourcing could be one way to combat the security concern, it also makes the task of computation over encrypted data in general a very difficult problem. Second, since the operational details inside the cloud are not transparent enough to customers, no guarantee is provided on the quality of the computed results from the cloud. ., to protect the sensitive input and output data and to validate the computation result integrity, it would be hard to expect customers to turn over control of their computing needs from local machines to cloud solely based on its economic savings. Focusing on the engineering and scientific computing problems, this paper 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. By interior point methods, system optimization problems can be converted to a system of nonlinear equations, which is then solved as a sequence of systems of linear equations. The analysis from existing approaches and the computational practicality motivates us to design secure mechanism of outsourcing LE via a completely different approach—iterative method, where the solution is extracted via finding successive approximations to the solution until the required accuracy is obtained. Compared to direct method, iterative method only demands relatively simpler matrix-vector operations with $O(n^2)$ computational cost, which is much easier to implement in practice and widely adopted for large-scale LE. To the best of our knowledge, no existing work has ever successfully tackled secure protocols for iterative methods on solving large-scale systems of LE in the computation outsourcing model, and we give the first study in this paper. Specifically, our mechanism utilizes the additive homomorphic encryption scheme to securely harness the cloud for finding successive approximations to the solution in a privacy-preserving and cheating-resilient manner.

IJERT

II. SYSTEM ARCHITECTURE

We consider a computation outsourcing architecture involving cloud customer and cloud server illustrated in above Figure. The customer has a large-scale LE problem Ax =b, denoted as $\emptyset = (a, b)$, to be solved. However, due to the lack of computing resources, he cannot carry out such expensive computation locally. Thus, the customer resorts to cloud server for solving the LE problem. Cloud customer would initialize a randomized key generation algorithm and prepare
the LE problem into some encrypted form_K via key K. Transformation and encryption operations will be needed when necessary. Then the cloud customer would use the encrypted form_K of LE to start the computation outsourcing process. In case of using iterative methods, the protocol ends when the solution within the required accuracy if found. Finally the cloud customer would verify the encrypted result produced from cloud server, using randomized secret key K. A correct output x to the problem is produced by decrypting the encrypted output. When the fails the customer outputs indicating the cloud server was cheating.

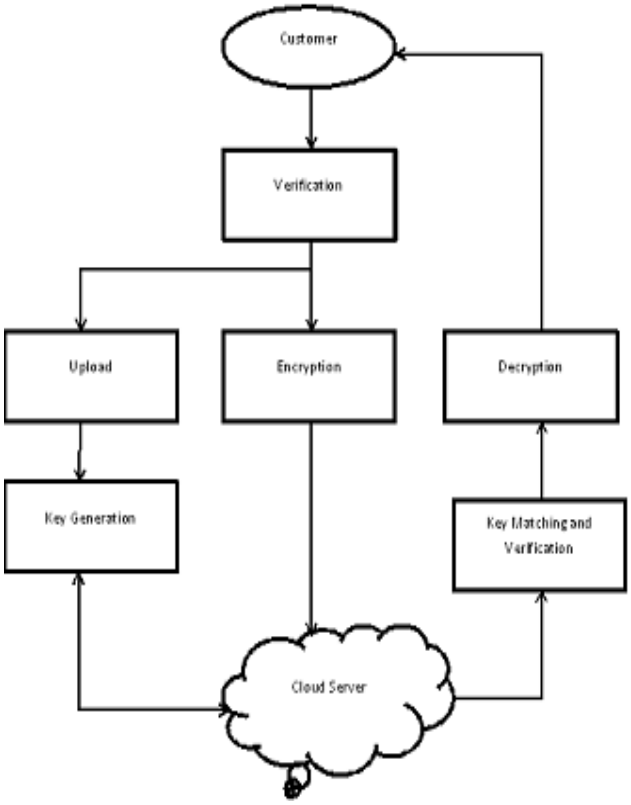

Fig. 1.Architecture of Secure Outsourcing Linear Equations In Cloud Computing

A. Data Flow Diagram

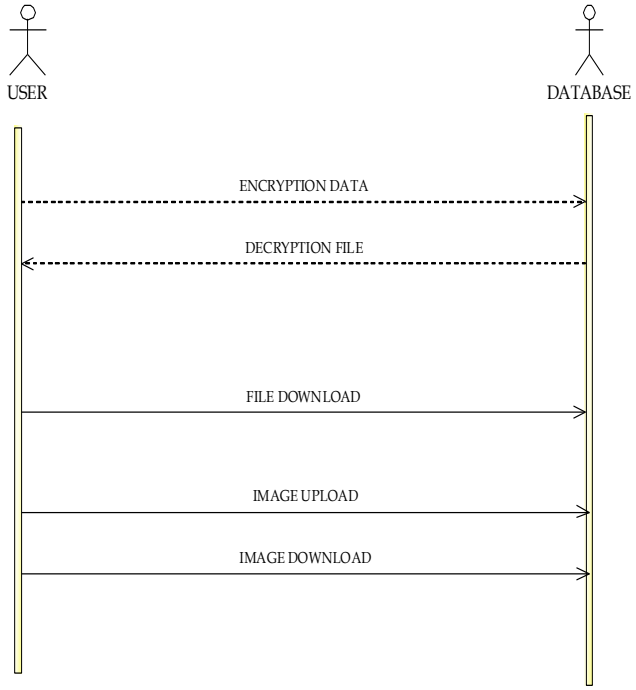

Fig. 2.Data Flow Diagram For Securely Outsourcing Linear Equations

In the data flow diagram, The customer has a large-scale LE problem s to be solved. However, due to the lack of computing resources, the cannot carry out such expensive computation locally. Thus, the customer resorts to cloud server for solving the LE problem. Cloud customer would initialize a randomized key generation algorithm and prepare the LE problem into some encrypted form_K via key K. Transformation and encryption operations will be needed when necessary. Then the cloud customer would use the encrypted form_K of LE to start the computation outsourcing process. In case of using iterative methods, the protocol ends when the solution within the required accuracy if found. Finally the cloud customer would verify the encrypted result produced from cloud server, using randomized secret key K. A correct output x to the problem is produced by decrypting the encrypted output. When the fails the customer outputs indicating the cloud server was cheating. Customer is going to upload and download image, file etc.

III. THE PROPOSED SCHEME

We propose a very efficient cheating detection mechanism to effectively verify in one batch of all the computation results by the cloud server from previous algorithm iterations with high probability. We formulate the problem in the computation outsourcing model for securely solving large-scale systems of LE via iterative methods, and provide the secure mechanism design which fulfills input/output privacy, cheating resilience, and efficiency. Our mechanism brings computational savings as it only incurs O(n) local computation burden for the customer within each algorithm iteration and demands no unrealistic IO cost, while solving large scale LE locally usually demands more than O(n2) computation cost in terms of both time and memory requirements. We explore the algebraic property of matrix-vector multiplication to design a batch result verification mechanism, which allows customers to verify all answers computed by cloud from previous iterations in one batch, and further ensures both the efficiency advantage and the robustness of the design. 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.

IV. TECHNIQUES

A. Pseudo-Random Key Generator

PRKG is a program written for, and used in, probability and statistics applications when large quantities of random digits are needed. Most of these programs produce endless strings of single-digit numbers, usually in base 10, known as the decimal system. When large samples of pseudo-random numbers are taken, each of the 10 digits in the set {0,1,2,3,4,5,6,7,8,9} occurs with equal frequency, even though they are not evenly distributed in the sequence. Many algorithm s have been developed in an attempt to produce truly random sequences of numbers, endless strings of digits in which it is theoretically impossible to predict the next digit in the sequence based on the digits up to a given point. But the very existence of the algorithm, no matter how sophisticated, means that the next digit can be predicted! This has given rise to the term pseudo-random for such machine-generated strings of digits. They are equivalent to random-number sequences for most

IJERT

applications, but they are not truly random according to the rigorous definition. The digits in the decimal expansions of irrational number s such as pi (the ratio of a circle's circumference to its diameter in a Euclidean plane), *e* (the natural- logarithm base), or the square roots of numbers that are not perfect squares (such as $2^{1/2}$ or $10^{1/2}$) are believed by some mathematicians to be truly random. But computers can be programmed to expand such numbers to thousands, millions, billions, or trillions of decimal places; sequences can be selected that begin with digits far to the right of the decimal (radix) point, or that use every second, third, fourth, or *n* th digit. However, again, the existence of an algorithm to determine the digits in such numbers is used by some theoreticians to argue that even these single-digit number sequences are pseudo-random, and not truly random. The question then becomes, is the algorithm accurate (that is, random) to infinity, or not? -- And because no one can answer such a question definitively because it is impossible to travel to infinity and find out, the matter becomes philosophical.

B. Iterative method

In many engineering computing and industrial applications, iterative method has been widely used in practice for solving large-scale LE and sometimes is the mandatory choice over direct method due to its ease of implementation and relatively less computational power consumption, including the memory and storage IO requirement. We now review some basics on the general form of stationary iterative methods for solving LE problems. A system of linear equations is written as

$$Ax = b; \qquad (1)$$

Where x is the n * 1 vector of unknowns, A is an n * n (nonsingular) coefficient matrix, and b is an n *1 right-hand side vector (so called constant terms). Most iterative methods involve passing from one iteration to the next by modifying a few components of some approximate vector solution at a time until the required accuracy is obtained. Without loss of generality, we focus on Jacobi iteration here and throughout the paper presentation for its simplicity. Though extensions to other stationary iterative methods can be possible, we don't study them in the current work. We begin with the decomposition: A =D+ R, where D is the diagonal component and R is the remaining matrix. Then, the (1) can be written as Ax = (D + R) x = b, and finally reorganized as: $x = -D^{-1}. R. x + D^{-1}. b$. According to the Jacobi method, we can use an iterative technique to solve the left hand side of this expression for $x^{(k+1)}$, using previous value for $x^{(k)}$ on the right hand side. If we denote iteration matrix $T = -D^{-1}. R$ and $c = D^{-1}. b$, the above iterative equations can be represented as

$$X^{(k+1)} = T.x^{(k)} + c. \qquad (2)$$

It is the case for a large body of LE problems derived from many real-world applications.

C. Homomorphic Encryption

Our construction utilizes a semantically secure encryption scheme with additive homomorphic property. Given two Integers x_1 and x_2, we have $Enc(x_1 + x_2) = Enc(x_1) * Enc(x_2)$, and also $Enc(x_1 \_ x_2) = Enc(x_1)^{x2}$. In our implementation we adopt the Paillier cryptosystem. For a vector $x = (x_1, x_2. \ . \ . \ Xn)^T \pounds (Z_N)$ n, we use Enc(x) to denote the coordinate-wise encryption of x: $Enc(x) = (Enc(x_1), Enc(x_2) \ . \ . \ . \ Enc(x_n))^T$. For some n * n matrix T, where each of the component T[i, j] in T is from Z_N , we denote the component-wise encryption of T as Enc(T), and we have Enc(T)[i, j]= Enc(T[i, j]).

V. RESULTS

Fig.3 HOME PAGE

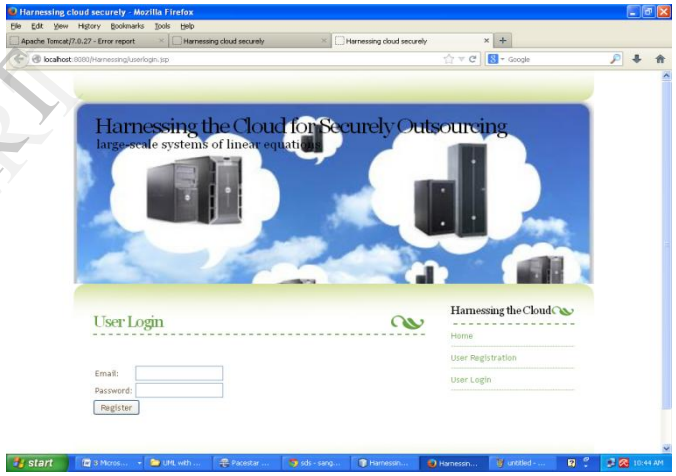

Fig.4 USER LOGIN PAGE

Fig.5 USER REGISTRATION

IJERT

Fig.6 IMAGE SEARCH

Fig.7 FILE SEARCH

VI. RELATED WORK

Recently, a general result of secure computation outsourcing has been shown viable in theory [18], which is based on Yao's garbled circuits and Gentry's fully homomorphic encryption (FHE) scheme. However, applying this general mechanism to our daily computations would be far from practical, due to the extremely high complexity of FHE operation and the pessimistic circuit sizes that can hardly be handled in practice. Instead of outsourcing general functions, in the security community, Atallahetal. Explore a list of customized solutions for securely out-sourcing specific computations. In, they give the first investigation of secure outsourcing of numerical and scientific computation, including LE. Though a set of problem dependent disguising techniques are proposed, they explicitly allow private information leakage. Besides, the important case of result verification is not considered. In, Atallah and Bejanmin give a protocol design for secure matrix multiplication outsourcing. The design is built upon the assumption of two no colluding servers and thus vulnerable to colluding attacks. Later on in, Atallah and Frikken give an improved protocol for secure outsourcing matrix multiplications based on secret sharing, which outperforms their previous work in terms of single server assumption and computation efficiency. But the drawback is that due to secret sharing technique, all scalar operations in original matrix multiplication are expanded to polynomials, introducing significant communication over-head. Considering the case of the result verification, the communication overhead must be further doubled, due to the introducing of additional precomputed "random noise" matrices. In short, these solutions, although elegant, are still not efficient enough for immediate practical uses on large-scale problems, which we aim to address for the secure LE outsourcing in this paper. Wang et al. Give the first study of secure outsourcing of linear programming in cloud computing. Their solution is based on problem transformation, and has the advantage of bringing customer savings without introducing substantial overhead on cloud. How-ever, those techniques involve cubic-time computational burden matrix-matrix operations, which may not be handled by the weak customer in our assumption. Very recently, Blanton et al. explored secure outsourcing all-pair distance calculations of large-scale biometric data. Their focus is on result verification, which leverages certain structures of the distance computations and the framework of adding fake items and random sampling.

IJERT

VII. CONCLUSIONS

In this paper, we investigated the problem of securely outsourcing large-scale LE in cloud computing. Different from previous study, the computation outsourcing frame-work is based on iterative methods. In particular, the design requires a one-time amortizable setup phase with $O(n^2)$ cost, and then each following iterative algorithm execution only incurs O(n) local computational cost with the benefits of easy-to-implement and less memory requirement in practice. We also investigated the algebraic property of the matrix-vector multiplication and developed an efficient and effective cheating detection scheme for robust result verification. Thorough security analysis and extensive experiments on the real cloud platform demonstrate the validity and practicality of the proposed mechanism.

VIII. REFERENCES

[1] C. Wang, K. Ren, J. Wang, and K. Mahendra Raje Urs,"Harnessing the Cloud for Securely Solving Large-Scale Systems of Linear equations," Proc. 31st Int'l Conf. Distributed Computing Systems (ICDCS), pp. 549-558, 2011.

[2] K. Forsman, W. Gropp, L. Kettunen, D. Levine, and J. Salonen,"Solution of Dense Systems of Linear Equations Arising from Integral-equation Formulations," IEEE Antennas and Propagation Magazine, vol. 37, no. 6, pp. 96-100, Dec. 1995.

[3] A. Edelman, "Large Dense Numerical Linear Algebra in 1993: The Parallel Computing Influence," Int'l J. High Performance Computing Applications, vol. 7, no. 2, pp. 113-128, 1993.

[4] V. Prakash, S. Kwon, and R. Mittra, "An Efficient Solution of a Dense System of Linear Equations Arising in the Method-of-Moments Formulation," Microwave and Optical Technology Letters,vol. 33, no. 3, pp. 196-200, 2002.

[5] B. Carpentieri, "Sparse Preconditioners for Dense Linear Systems from Electromagnetic Applications," PhD dissertation, CERFACS, Toulouse, France, 2002.

[6] R. Cramer and I. Damgard, "Secure Distributed Linear Algebra in a Constant Number of Rounds," CRYPTO: Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology, 2001.

[7] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS), pp. 253-262, 2010.
[8] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable,and Fine-Grained Access Control in Cloud Computing," Proc. IEEE INFOCOM, pp. 534-542, 2010.
[9] C. Wang, K. Ren, S. Yu, and K. Mahendra Raje Urns, "Achieving Usable and Privacy-Assured Similarity Search Over Outsourced Cloud Data," Proc. IEEE INFOCOM, pp. 451-459, 2012.
[10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, second ed. MIT press, 2008.
[11] D. Benjamin and M.J. Atallah, "Private and Cheating-Free Outsourcing of Algebraic Computations," Proc. Sixth Conf. Privacy, Security, and Trust (PST), pp. 240-245, 2008.
[12] M. Atallah and K. Frikken, "Securely Outsourcing Linear Algebra Computations," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 48-59, 2010.
[13] G. Dahlquist and A. Bjorck, Numerical Methods. Dover Publications, 2003.
[14] M. Bellare, J. Garay, and T. Rabin, "Fast Batch Verification for Modular Exponentiation and Digital Signatures," Eurocrypt: Proc. Int'l Conf. the Theory and Application of Cryptographic Techniques, pp. 236-250, 1998.
[15] J. Camenisch, S. Hohenberger, and M. Pedersen, "Batch Verification of Short Signatures," EUROCRYPT: Proc. 26th Ann. Int'l Conf. Advances in Cryptology, pp. 243-263, 2007.

IJERT