Secure Outsourcing Algorithm of Polynomials in Cloud Computing

Xianlin Zhou
Collage of Mathematics
and Software Science,
Sichuan Normal University,
Chengdu, Sichuan, P. R. China
flyinufo@qq.com

Yong Ding, Zhao Wang, Xiumin Li
School of Computer Science and Information Security,
Guilin University of Electronic Technology,
Guilin, Guangxi, P. R. China
stone_dingy@126.com,824420896@qq.com,
630717447@qq.com

Zheng Xu *
The Third Research Institute of
the Ministry of Public Security,
Shanghai, China
zhengxu@shu.edu.cn

Jun Ye *
Sichuan University of Science & Engineering,
Artificial Intelligence Key
Laboratory of Sichuan Province,
Zigong, Sichuan, P. R. China
yejun@suse.edu.cn

Abstract— **In the era of information explosion, people have to deal with huge amount of data. It is a great computation burden for the resource-constrained clients. Cloud computing connects large amounts of network resources, and forms a vast pool of resources. It provides much convenience for people. Clients can outsource the complex computation task to the powerful cloud server. In this way, the computation burden of clients can be greatly reduced. In this paper a new algorithm of secure outsourcing for polynomials is proposed. In the computation process, the computation polynomial is hidden to cloud server, and the inputs and outputs of polynomials will not revealed. In addition, clients can verify the result easily.**

Keywords– Secure Outsourcing; Cloud Computing; Polynomial

1 Introduction

With the rapid development of society, we have entered an era of information explosion. People have to face the huge amount of data. It brings much trouble to handle the complex computation task with huge data.

Cloud computing [19, 10, 22, 17, 5] shares the computing resources, which provides a lot of convenience for the resource-constrained clients. Clients can outsource the heavy computation work to cloud server with pay-per-use manner. And the computation cost of clients can be greatly reduced. Especially in big data era, cloud computing is a perfect technique, which helps people deal with big data [15, 12, 14]. The high reliability, strong processing capacity, large storage space of cloud computing make the clients with limited storage capacity and computing power to remotely operate the big data [17, 18]. The framework of big data based on cloud computing, realize a distributed operating system by utilizing the powerful computation and storage. It provides the efficient data access [16, 21]. Although there are many advantages, the development and application of cloud computing are seriously constrained by the data security and privacy issues. In cloud computing, the resource-constrained clients conveniently obtain the results from the cloud servers. However, it is hard to find a reliable cloud service provider in cloud computing. The cloud server may want to obtain the inputs and outputs of clients. In order to get more benefits and save the resources, cloud server will probably return a random result without computing. And the wrong results would be returned due to the loophole of software and the fault of hardware. Although some solutions in this aspect are studied, the verification is unsatisfactory. This forms the notion of verifiable computation [7, 20, 13, 4, 9], which enables clients can check the correctness of the returned results.

In 2002, secure outsourcing for scientific computing is studied by Atallah et al. [1]. They pro-

DOI reference number: 10.18293/SEKE2016-022

posed some camouflage technologies to protect the privacy of the outsourced computations. However, these does not solve the problem of the verifiability for computing results. In 2005, Hohenberger and Lysyanskaya [11] proposed the formal security definition of outsourcing. In 2008, Benjamin and Atallah [3] used homomorphic encryption to construct a secure outsourcing scheme for linear algebraic computation, in which the client can verify the results. And in 2009, based on ideal lattices, a fully homomorphic encryption scheme was proposed by Gentry [8]. However, the efficiency is low.

Polynomials are often used in many application fields, such as, signal processing, data analysis etc. In 2011, Benabbas et al. [2] proposed an algorithm of secure outourcing for polynomials based on homomorphic encryption. In 2012, Fiore and Gennaro [6] proposed a scheme for verifiable delegation of large polynomials. However, in these two schemes, the inputs would be revealed. In 2016, Ye et al. [20] proposed a scheme for secure outsourcing polynomials, in which an extra polynomial will be outourced for verification.

Our Contributions The main contributions are as follows, the transformation technique and the secure scheme for secure outsourcing of polynomials. The transformation technique guarantees the security of the outsourcing polynomial, the cloud server cannot get the real polynomial which will be computed. In the secure outsourcing scheme, the inputs and outputs are keeping privacy, and client can easily verify the returned results.

1.1 Organization of this Paper

The organization of this paper is as follows. Some preliminaries are given in Section 2. The algorithm of secure outsourcing for polynomials is given in Section 3. Then in Section 4 we give the security analysis. Finally, the conclusion is made in Section 5.

2 Preliminaries

Outsource-security: An algorithm is said to be an outsource-secure algorithm if:

Correctness. The result returned from the cloud server is the correct implementation of the algorithm.

Security. For all probabilistic polynomial-time adversary, the original computation cannot be obtained from the outsourced disguised computation.

Verifiable Oursourcing Computation: We follow the definition in [20].

A verifiable outsourcing computation scheme is defined by the following algorithms:

KeyGen$(f,k) \rightarrow (PK,SK)$: Based on the security parameter k, the key generation algorithm generates a key pair (PK,SK) for the function f. PK is provided to the server, and client keeps SK.

ProGen$(x) \rightarrow (\sigma_x, V_x)$: The problem generation algorithm is run by client, who uses SK to encode the input x as σ_x which is given to server, and a verification key V_x which is kept private by client.

Compute$(\sigma_x) \rightarrow (\sigma_y)$: The algorithm is run by the server to compute an encoded version of the output σ_y.

Verify$(V_x, \sigma_y) \rightarrow (y \cup \perp)$: The algorithm returns the value $y = f(x)$ or $\perp$ indicating that σ_y does not equal to $f(x)$.

A verifiable computation scheme should be correct, secure and efficient.

3 Secure Outsourcing of Polynomials

We consider the following scenario. A resource-constrained client wants to outsource a high degree polynomial with fixed coefficients. This polynomial will be used latter for some applications frequently. The polynomial, the inputs and the outputs should be blind to cloud server. And the client should verify the correctness of the result efficiently.

3.1 Design Goals

To securely outsourcing the computation of polynomials efficiently, the design goals of our system are as follows.

- **Disguise**. To design a transformation method which disguise the real computing polynomials, so that the cloud server cannot get the information of original polynomials.

- **Privacy Preserving**. To prevent the cloud server from learning the information of the inputs and outputs.

- **Verification**. To make sure that the server returns the correct computing results.

- **Efficiency**. The computation cost of verification should be greatly less than that of polynomial computation.

3.2 System Model

There are two parties in the model, a resource-constrained client and a untrusted powerful cloud server. The system model is shown in Fig. 1.

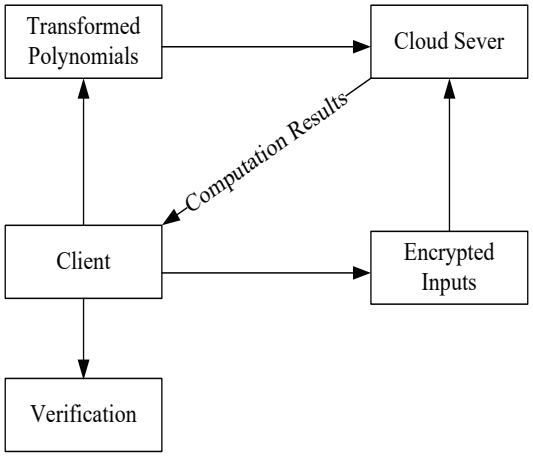

Figure 1: System Model

Client firstly transform the original polynomial into a disguised polynomial, and outsources the disguised polynomial to the cloud server. When client wants to do some computation on the polynomial, he/she outsoutces the encrypted inputs to the cloud servers. After the computing, cloud server returns the computation results to the client. Then client verify the computation results. If the returned results are correct, client will transform the returned results into the real computation results.

In the following we give the polynomial transformation technique and generate our outsourcing method based on the technique in [2].

3.3 Transformation Technique

The polynomial

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

where $a_i \in \mathbb{Z}_p, 0 \leq i \leq n$ is a high degree polynomial, will be outsourced to cloud server. Client want to compute the function on the value of x.

For the secure outsourcing and efficient verification, a transformed polynomial $F(\sigma_x)$ is constructed.

Client selects $r \in_R \mathbb{Z}_p$, $c \in_R \mathbb{Z}_p$ and $d \in_R \mathbb{Z}_p$, and computes

$$b_0 = c + a_0.$$

The coefficients of transformed polynomial

$$F(\sigma_x) = b_0 + b_1 \sigma_x + b_2 \sigma_x^2 + \cdots + b_n \sigma_x^n$$

is generated as

$$b_i = a_i r^i - d^i$$

where $1 \leq i \leq n$. $F(\sigma_x)$ will be outsoutced to cloud server.

3.4 Our Scheme

We assume σ_x is an encoded input and σ_y is an encoded output, the polynomial F is a encrypted function.

Client wants to computes

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \mod p$$

where $f(x)$ is a high degree polynomial. He/she firstly transforms $f(x)$ into $F(\sigma_x)$ by using the transform technique, and then delegates it to the cloud server. And the client can verify the correctness of the result.

Initialization. Client randomly selects six numbers r, c, d, R, k_0 and k_1, then client generates

$$F(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n$$

where

$$b_0 = a_0 + c$$

and

$$b_i = a_i r^i - d^i$$

$i = 1, 2, \cdots, n$.

Then client computes

$$t_0 = g^{k_0 + Rb_0}$$

$$t_1 = g^{k_0 + k_1^i + Rb_i}$$

where $i = 1, 2, \cdots, n$.

Delegation. We denote $t = (t_0, t_1, \cdots, t_n)$. Client sends the polynomial

$$F(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n$$

and

$$t = (t_0, t_1, \cdots, t_n)$$

to cloud server.

When client wants to compute $f(x)$, client computes $\sigma_x = \frac{x}{r}$ and sends σ_x to cloud server. Then client computes

$$Z = \prod_{i=0}^{n} t_1^{\sigma_x^i} = g^{k_0 \frac{1-(k_1\sigma_x)^{n+1}}{1-k_1\sigma_x}}.$$

Computation. Cloud server computes

$$\sigma_y = F(\sigma_x)$$

and

$$T = \prod_{i=0}^{n} t_1^{\sigma_x^i}.$$

Then cloud server sends (σ_y, T) to client.

Verification. Client verifies whether following equation holds

$$T = Z g^{R\sigma_y}.$$

If not, the server gives the wrong answer, σ_y is not correct. If the equation holds, client can get the final result by computing

$$y = \sigma_y - y_1 \mod p$$

where

$$\begin{aligned} y_1 &= \sum_{i=1}^{n} d^i \sigma_x^i - c \\ &= \frac{d\sigma_x - (d\sigma_x)^{n+1}}{1 - d\sigma_x} - c. \end{aligned}$$

4 Security Analysis

Theorem 1. *The input and output of the polynomial are secure.*

Proof. The real input is x, however, the encrypted input is σ_x, where $\sigma_x = \frac{x}{r}$. For r is randomly chosen, the input x is keeping privacy.

The output client needs is $y = \sigma_y - y_1$. Cloud server can get σ_y, however, it cannot get y_1.

For

$$y_1 = \frac{d\sigma_x - (d\sigma_x)^{n+1}}{1 - d\sigma_x} - c$$

where d and c are randomly chosen by client. Cloud server can just get σ_x. And in the computation process, cloud server cannot get the private parameters d and c from the outsourced polynomial.

Hence, the input and output would not revealed. □

5 Conclusion

In big data era, people cannot afford the more and more complex computation work due to the constrained computation resources. Outsourcing computation helps people to solve the heavy computation task. In this paper, a new algorithm for secure outsourcing of high degree polynomials is given. And we introduce the transformation technique, with which the real polynomial will be hidden to the untrusted cloud server. In addition, the input and output will not be revealed in the computation process. Finally, the clients can easily verify the returned result.

ACKNOWLEDGMENT

This work was supported in part by the Science Founding of Artificial Intelligence Key Laboratory of Sichuan Province (2014RYJ06), in part by the Scientific Research Fund Project of Sichuan Normal University (15YB008), in part by the Guangxi experiment center of information science Foundation, in part by the Innovation Project of Guangxi Graduate Education(XJYC2012020), in part by the National Science and Technology Major Project

under Grant (2013ZX01033002-003), in part by the National High Technology Research and Development Program of China (863 Program) under Grant (2013AA014601), in part by the National Science Foundation of China under Grant (61300202), and in part by the Science Foundation of Shanghai under Grant (13ZR1452900).

References

[1] M.J. Atallah, K.N. Pantazopoulos, J.R. Rice, and E.E. Spafford. Secure outsourcing of scientific computations. *Advances in Computers*, 54:215–272, 2002.

[2] S. Benabbas, R. Gennaro, and Y. Vahlis. Verifiable delegation of computation over large datasets. In *Advances in Cryptology–CRYPTO 2011*, pages 111–131. Springer, 2011.

[3] D. Benjamin and M.J. Atallah. Private and cheating-free outsourcing of algebraic computations. In *Sixth Annual Conference on Privacy, Security and Trust, PST 2008, Fredericton, New Brunswick, Canada*, pages 240–245. Springer-Verlag, October 2008.

[4] S.G. Choi, J. Katz, R. Kumaresan, and C. Cid. Multi-client non-interactive verifiable computation. In *Proc. of the 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan*, volume 7785, pages 499–518. Springer-Verlag, March 2013.

[5] S. El-Sayed, H. Kader, M. Hadhoud, and D. AbdElminaam. Mobile cloud computing framework for elastic partitioned/modularized applications mobility. *International Journal of Electronics and Information Engineering*, 1(2):53–63, 2014.

[6] D. Fiore and R. Gennaro. Publicly verifiable delegation of large polynomials and matrix computations, with applications. In *Proc. of the 2012 ACM conference on Computer and communications security, ACM New York, NY, USA*, pages 501–512. Springer-Verlag, October 2012.

[7] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Advances in Cryptology–CRYPTO 2010*, volume 6223, pages 465–482. Springer, 2010.

[8] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[9] S.D. Gordon, J. Katz, F. Liu, E. Shi, and H. Zhou. Multi-client verifiable computation with stronger security guarantees. In *Theory of Cryptography*, pages 144–168. Springer, 2015.

[10] I.A.T. Hashem, I. Yaqoob, N.B. Anuar, S. Mokhtar, A. Gani, and S.U. Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

[11] S. Hohenberger and A. Lysyanskaya. How to securely outsource cryptographic computations. In *Proc. of the second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA*, volume 3378, pages 264–282. Springer-Verlag, February 2005.

[12] C Hu, Z. Xu, Y. Liu, L. Mei, L. Chen, and X. Luo. Semantic link network-based model for organizing multimedia big data. *IEEE Transactions on Emerging Topics in Computing*, 2(3):376–387, 2014.

[13] B. Parno, J. Howell, C. Gentry, and M. Raykova. Pinocchio: Nearly practical verifiable computation. In *Proc. of the 2013 IEEE Symposium on Security and Privacy, IEEE Computer Society Washington, DC, USA*, pages 238–252. Springer-Verlag, March 2013.

[14] Z. Xu, Y. Liu, L. Mei, C. Hu, and L. Chen. Semantic based representing and organizing surveillance big data using video structural description technology. *Journal of Systems and Software*, 102:217–225, 2015.

[15] Z. Xu, Y. Liu, N. Yen, L. Mei, X. Luo, X. Wei, and C. Hu. Crowdsourcing based description of urban emergency events using social media big data. *IEEE Transactions on Cloud Computing*, 2016. DOI: 10.1109/TCC.2016.2517638.

[16] C.g Yang and J. Ye. Secure and efficient fine-grained data access control scheme in cloud computing. *Journal of High Speed Networks*, 21(4):259–271, 2015.

[17] J. Ye, X. Chen, and J. Ma. An improved algorithm for secure outsourcing of modular exponentiations. In *Proc. of 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 24-27, March*, pages 73–76. IEEE, 2015.

[18] J. Ye, M. Miao, P. Chen, and X. Chen. Request-based comparable encryption scheme with multiple users. In *10th International Conference on Broadband and Wireless Computing,Communication and Applications, BWCCA 2015, Krakow, Poland, November 4-6, 2015*, pages

414–416, 2015.

[19] J. Ye and J. Wang. Secure outsourcing of modular exponentiation with single untrusted server. In *Proc. of 18th International Conference on Network-Based Information Systems (NBiS), 2-4, September*, pages 643–645. IEEE, 2015.

[20] J. Ye, H. Zhang, and C. Fu. Verifiable delegation of polynomials. *International Journal of Network Security*, 18(2):283–290, 2016.

[21] J. Ye, W. Zhang, S. Wu, Y. Gao, and J. Qiu. Attribute-based fine-grained access control with user revocation. In *Information and Communication Technology*, pages 586–595. Springer, 2014.

[22] D. Zissis and D. Lekkas. Addressing cloud computing security issues. *Future Generation Computer Systems*, 28(3):583–592, 2012.