Analysis of Two Outsourcing Algorithms for Solving Quadratic Congruence

Lihua Liu and Yujie Li
(Corresponding author: Lihua Liu)
Department of Mathematics, Shanghai Maritime University
Haigang Ave 1550, Shanghai, 201306, China
Email: liulh@shmtu.edu.cn
(Received Oct. 16, 2021; Revised and Accepted Mar. 23, 2022; First Online Apr. 11, 2022)

Abstract

We show that the outsourcing algorithms [IEEE ITJ, 7(4), 2020, 2968–2981] for solving quadratic congruence in the Internet of Things are flawed. (1) The Cipolla algorithm is unsuitable for the discussed scenario. The underlying modulus is generally a composite containing two strong primes to resist some factorization algorithms. Besides, the Rabin cryptosystem explicitly requires that $p \equiv q \equiv 3 \bmod 4$. In this case, the Cipolla algorithm is unnecessary. (2) The outsourcer can finish the computation solely, even if $p \not\equiv 3 \bmod 4$ and $q \not\equiv 3 \bmod 4$. He doesn't have to outsource the original problem because he must pay out equal-cost $O(\log^3 p)$ in the proposed outsourcing scenario.

Keywords: Adleman-Manders-Miller algorithm; Cipolla Algorithm; Pollard Method; Quadratic Congruence; Strong Prime

1 Introduction

The Internet of Things (IoT) consists of a large amount of resource constrained devices to collect and compute data. These devices need to execute some public-key cryptographic protocols for confidentiality and authentication [2]. But some public-key computations are too expensive for these devices to finish. It becomes usual for these resource constrained devices to outsource those heavy computations to cloud or edge servers.

In the outsourcing scenario, one has to tackle some security challenges [14]. Usually, it requires that:

- The sensitive information contained in the outsourced data should not be exposed to the cloud servers.
- The outsourcer can verify the correctness of the returned results.
- The outsourcer can save much computational cost in comparison with the incurred communication cost.

Dreier and Kerschbaum [10] have put forth a method for secure outsourcing of linear programming. After that, Wang *et al.* [21] proposed a scheme for outsourcing large-scale systems of linear equations. But its homomorphic encryption system [15] was not compatible with Jacobi iteration [3]. In 2014, Chen *et al.* [7] presented one algorithm for outsourcing linear regression problem, but neglected to check whether the client can solve the original problem solely [4].

In 2015, Salinas *et al.* [18, 19] have presented an outsourcing scheme for large-scale sparse linear systems of equations. In 2018, Ding *et al.* [9] pointed out that in the Salinas *et al.*'s scheme the cloud server can recover a client's input. In 2020, Cao and Markowitch [5] argued that in the discussed scenario it was unnecessary for a client to outsource the problem because he can finish the computations solely. Recently, Wang *et al.* [11, 22] have presented a survey for reversible data hiding for VQ-compressed images. Pan *et al.* [8, 12, 13, 16, 17, 20] put forth some batch verification schemes for identifying illegal signatures, smart card-based password authentication schemes, and data collaboration scheme with hierarchical attribute-based encryption in cloud computing scenario.

The Rabin cryptosystem is based on the intractability of solving $x^2 \equiv \bmod n$, where n is an RSA modulus. It has been proved that the security of Rabin cryptosystem was equivalent to factoring n, while the security of RSA has not yet been proven. So, in some cases the Rabin cryptosystem is more appreciated because the encryptor only needs to do one multiplication modulo n. For example, the IoT security framework makes use of the Rabin encryption to offer confidentiality.

Table 1: Cipolla algorithm

Let p be an odd prime, $n \in \mathbb{F}_p$ be a quadratic residue.
Find $a \in \mathbb{F}_p$ such that $(a^2 - n)^{\frac{p-1}{2}} \equiv -1 \bmod p$. Compute the quadratic root $x = (a + \sqrt{w})^{\frac{p+1}{2}} \bmod p$ within the filed $F_p(\sqrt{w})$, where $w = a^2 - n$.

The Cipolla algorithm (Table 1) can be used to solve quadratic congruences. Recently, Zhang *et al.* [23] have presented two outsourcing algorithms based on Cipolla algorithm. In this note, we show that the outsourcing algorithms have two flaws.

2 Review of the Algorithms

Given the odd prime p and $n \in \mathbb{F}_p$, the client transforms the two numbers into $p' = pq, n' = n - r_1 p$, where q, r_1 are two random blinders. We now only describe the second outsourcing algorithm as below (Table 2). Its correctness is based on that

$$\begin{aligned} x \equiv & (a + \sqrt{a^2 - n'})^k R'_2 \bmod p \\ \equiv & (a + \sqrt{a^2 - n'})^k ((a + \sqrt{a^2 - n'})^{d'_2} \bmod p') \bmod p \\ \equiv & (a + \sqrt{a^2 - n'})^{k + d'_2} \bmod p \\ \equiv & (a + \sqrt{a^2 - n'})^{(p+1)/2} \bmod p \\ & ((a^2 - n')^{d'} \bmod p') \bmod p \\ & \equiv (a^2 - n')^{(p-1)/2 + r_2(p-1)} \bmod p \\ & \equiv (a^2 - n')^{(p-1)/2} \bmod p \\ & \equiv (a^2 - n)^{(p-1)/2} \bmod p \equiv -1. \end{aligned}$$

3 Analysis

In general, an outsourcing algorithm should meet two basic requirements: privacy—nobody can know the client's input and output except himself; efficiency—the client can save much cost in comparison with solving the original problem solely. But we find the proposed outsourcing algorithms fail to meet the second requirement.

3.1 Strong Primes Should Be Chosen

In order to resist some factorization algorithms such as the Pollard $\rho + 1$ or $\rho - 1$ methods, PKCS suggests the using of strong primes. A strong prime p satisfies that $p - 1$ contains a large prime factor, and $p + 1$ also contains a large prime factor. The Rabin cryptosystem in particular requires that $p \equiv q \equiv 3 \bmod 4$. In this case, the user can simply recover the square root by computing $n^{\frac{p+1}{4}} \bmod p$. The claim that [page 2979, Ref. [23]] "in the Rabin cryptosystem, p and q are not necessary to be $p \equiv q \equiv 3 \bmod 4$" is incorrect. To the best of our knowledge, the Cipolla algorithm is unnecessary for a public key cryptographic scheme based on the intractability of factorization.

3.2 The Outsourcer Can Finish The Computation Solely

The outsourcing algorithms fail to save much cost for the outsourcer, even if $p \not\equiv 3 \bmod 4$. As we see, the outsourced task is just to do the computation of finding a such that

$$(a^2 - n)^{\frac{p-1}{2}} \equiv -1 \bmod p$$

For a randomly chosen $a \in \mathbb{F}_p$, the checking requires $O(\log^2 p)$ cost by computing the Legendre Symbol $\left(\frac{a^2-n}{p}\right)$. Assuming Extended Riemann Hypothesis, it requires $O(\log p)$ tests [1] to find out a quadratic nonresidue ρ such that $\rho^{\frac{p-1}{2}} \equiv -1 \bmod p$. The modular exponentiation

$$(a + \sqrt{w})^{\frac{p+1}{2}} \bmod p.$$

requires $O(\log^3 p)$ cost. Thus, the client only needs to pay $O(\log^3 p)$ cost to find the square root, if he tries to solve the problem solely.

In the outsourcing scenario the client needs to pay $O(\log k \log^2 p)$ cost to compute

$$(a + \sqrt{a^2 - n'})^k R'_2 \bmod p.$$

The cloud (not knowing the modulus p) should ultimately find out $a \in \mathbb{F}_{p'}$ such that

$$((a^2 - n')^{d'} \bmod p') \bmod p \equiv -1.$$

The procedure needs to do $O(\log p)$ interactions between the client and the cloud, which requires much cost because the communicators and transferred data should be authenticated. Usually, it requires at least $O(\log^2 p)$ cost (equal to doing a multiplication of two integers with the same length $\log p$) for the client to authenticate the transferred data in each loop. So, the client needs to pay out $O(\log^3 p)$ cost at least. See Table 3 for the cost comparisons. Thus, it is unnecessary for the client to outsource the original problem because he needs to pay out equal cost $O(\log^3 p)$ in the outsourcing scenario.

Other flaws. The communication cost analysis (Table II, [23]) neglects the cost for underlying communicators authentication and data integrity authentication. The listed references [16, 21] are misleading due to the shortcomings shown in [3, 5]. By the way, the expressions $R'_1 = (a^2 - n)^{d'}$ and $R'_1 = a^2 - n'$ (page 2975, Ref. [23]) are not computed over the ring $\mathbb{Z}_n$, which should be revised as $R'_1 = (a^2 - n)^{d'} \bmod p'$, where p' is a secret prime factor owned only by the outsourcer.

3.3 Further Discussions

There is a more efficient algorithm for root extraction, i.e., the Adleman-Manders-Miller algorithm [1] (Table 4). Its basic idea can be described as below. Write $p - 1 = 2^t s, 2 \nmid s$. Given a quadratic residue δ and a quadratic nonresidue ρ, i.e.,

$$(\delta^s)^{2^{t-1}} \equiv 1 \bmod p, \quad (\rho^s)^{2^{t-1}} \equiv -1 \bmod p.$$

If $t \geq 2$, then $(\delta^s)^{2^{t-2}} \bmod p \in \{1, -1\}$. Take $k_1 \in \{0, 1\}$ such that

$$(\delta^s)^{2^{t-2}} (\rho^s)^{2^{t-1} \cdot k_1} \equiv 1 \bmod p.$$

Table 2: SoSQC2

Client: $\{n, p\}$	Cloud
[Transformations] Pick $r_1, r_2 \in \mathbb{F}_p$, a short random integer k, and a large prime q. Compute $p' = pq, n' = n - r_1 p$, $d' = (p-1)/2 + r_2(p-1)$, $d'_2 = (p+1)/2 - k$. $\xrightarrow{n', d', d'_2, p'}$	
Check $R'_1 \equiv -1 \bmod p$. If it fails, ask for a new number until such an integer is found. $\xrightarrow{\text{OK}}$	[Quadratic nonresidue finding] Pick $a \in \mathbb{F}_p$, compute $R'_1 = (a^2 - n')^{d'} \bmod p'$. $\xleftarrow{R'_1}$ Upon receiving "OK", compute $R'_2 = (a + \sqrt{a^2 - n'})^{d'_2} \bmod p'$. $\xleftarrow{a, R'_2}$
[Retrieval] Compute $x = (a + \sqrt{a^2 - n'})^k R'_2 \bmod p$. Check that $x^2 \equiv n \bmod p$.	

Table 3: Cost comparisons

Unoutsourcing case	Outsourcing case
(1) Find $a \in \mathbb{F}_p$ such that $(a^2 - n)^{\frac{p-1}{2}} \equiv -1 \bmod p$, which requires $O(\log^3 p)$ cost.	(1) Do $O(\log p)$ interactions with the cloud to find $R'_1 \equiv -1 \bmod p$, which requires $O(\log^3 p)$ cost at least.
(2) Compute $(a + \sqrt{w})^{\frac{p+1}{2}} \bmod p$ which requires $O(\log^3 p)$ cost.	(2) Compute $(a + \sqrt{a^2 - n'})^k R'_2 \bmod p$ which requires $O(\log k \log^2 p)$ cost.

Since $(\delta^s)^{2^{t-3}} (\rho^s)^{2^{t-2} \cdot k_1} \bmod p \in \{1, -1\}$, take $k_2 \in \{0, 1\}$ such that

$$(\delta^s)^{2^{t-3}} (\rho^s)^{2^{t-2} \cdot k_1} (\rho^s)^{2^{t-1} \cdot k_2} \equiv 1 \bmod p.$$

Likewise, take $k_3, \cdots, k_{t-1} \in \{0, 1\}$ such that

$$\delta^s (\rho^s)^{2 \cdot k_1 + 2^2 \cdot k_2 + \cdots + 2^{t-1} \cdot k_{t-1}} \equiv 1 \bmod p.$$

Thus,

$$\left(\delta^{\frac{s+1}{2}}\right)^2 \left((\rho^s)^{k_1 + 2 \cdot k_2 + \cdots + 2^{t-2} \cdot k_{t-1}}\right)^2 \equiv \delta \bmod p.$$

Its computational complexity is $O(\log^3 p + t^2 \log^2 p)$ (see [6]). Since p is usually set as a strong prime, i.e., $t = 1$, it becomes $O(\log^3 p)$, and $\delta^{\frac{p+1}{2}} \equiv \delta \bmod p$.

If $4 \mid p + 1$, then $\delta^{\frac{p+1}{4}} \bmod p$ is just a square root of δ modulo p. This is the reason that the Rabin Cryptosystem specifies $p \equiv q \equiv 3 \bmod 4$. In this case, it avoids the need to find a quadratic nonresidue.

The complexities of Cipolla algorithm and Adleman-Manders-Miller algorithm are both dominated by the procedure to find a quadratic nonresidue. The Cipolla algorithm needs to find a special quadratic nonresidue which should be written as $a^2 - n$, while the Adleman-Manders-Miller algorithm only needs to find a common quadratic nonresidue ρ. So, the Adleman-Manders-Miller algorithm is more efficient than the Cipolla algorithm. Besides, the Adleman-Manders-Miller algorithm can be extended to r^{th} root extraction ($r > 2$).

Table 4: Adleman-Manders-Miller algorithm

Let p be an odd prime, $\delta \in \mathbb{F}_p$ be a quadratic residue.
Write $p - 1$ as $2^t s$, where $2 \nmid s$.
Find a quadratic nonresidue ρ, i.e., $\rho^{\frac{p-1}{2}} \equiv -1 \bmod p$.
Take $k_1, \cdots, k_{t-1} \in \{0, 1\}$, such that $\delta^s (\rho^s)^{2 \cdot k_1 + 2^2 \cdot k_2 + \cdots + 2^{t-1} \cdot k_{t-1}} \equiv 1 \bmod p$.
Compute the quadratic root $x \equiv \delta^{\frac{s+1}{2}} (\rho^s)^{k_1 + 2 \cdot k_2 + \cdots + 2^{t-2} \cdot k_{t-1}} \bmod p$.

4 Conclusion

We show that the Zhang *et al.*'s outsourcing algorithms cannot save much cost for the client. We want to stress that the Adleman-Manders-Miller algorithm is more efficient than the Cipolla algorithm because the latter needs to find out a special quadratic nonresidue.

Acknowledgements

We thank the National Natural Science Foundation of China (Project 61411146001). We are grateful to the reviewers for their valuable suggestions.

References

[1] L. Adleman, K. Manders, and G. Miller, "On taking roots in finite fields," in *Proceedings of 18th Annual Symposium on Foundations of Computer Science*, pp. 175–178. IEEE Computer Society, 1977.

[2] E. Bresson, D. Catalano, and D. Pointcheval, "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications," in *Proceedings of 9th International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2003*, pp. 37–54, Taipei, December 2003.

[3] Z. J. Cao and L. H. Liu, "Comment on 'harnessing the cloud for securely outsourcing large-scale systems of linear equations'," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1551–1552, 2016.

[4] Z. J. Cao, L. H. Liu, and O. Markowitch, "Comment on 'highly efficient linear regression outsourcing to a cloud'," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, p. 893, 2019.

[5] Z. J. Cao and O. Markowitch, "Comment on 'efficient secure outsourcing of large-scale sparse linear systems of equations'," *IEEE Transactions on Big Data*, 10.1109/TBDATA.2020.2995200.

[6] Z. J. Cao, Q. Sha, and X. Fan, "Adleman-Manders-Miller root extraction method revisited," in *Proceedings of 7th International Conference on Information Security and Cryptology*, pp. 77–85. Springer, 2011.

[7] F. Chen and *et al.*, "Highly efficient linear regression outsourcing to a cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 499–508, 2014.

[8] Y. H. Chen and *et al.*, "Research on the secure financial surveillance blockchain systems," *International Journal of Network Security*, vol. 22, no. 4, pp. 708–716, 2020.

[9] Q. Ding and *et al.*, "Efficient and secure outsourcing of large-scale linear system of equations," *IEEE Transactions on Cloud Computing*, 10.1109/TCC.2018.2880181.

[10] J. Dreier and F. Kerschbaum, "Practical privacy-preserving multiparty linear programming based on problem transformation," in *Proceedings of IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing*, pp. 916–924, Boston, USA, Oct. 2011.

[11] L. C. Huang, T. Y. Chang, and M. S. Hwang, "A conference key scheme based on the Diffie-Hellman key exchange," *International Journal of Network Security*, vol. 20, no. 6, pp. 1221–1226, 2018.

[12] L. H. Liu and *et al.*, "A note on one secure data self-destructing scheme in cloud computing," *International Journal of Network Security*, vol. 22, no. 1, pp. 36–40, 2020.

[13] L. H. Liu and L. M. Hong, "Analysis of one authenticated key agreement scheme for consumer usb mass storage devices resilient to unauthorized file decryption," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 10–16, 2021.

[14] D. Marinescu, *Cloud Computing Theory and Practice*. USA: Elsevier, 2013.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT 1999*, pp. 223–238, Prague, Czech Republic, May 1999.

[16] H. Pan and *et al.*, "Research on batch verification schemes for identifying illegal signatures," *International Journal of Network Security*, vol. 21, no. 6, pp. 1062–1070, 2019.

[17] H. T. Pan, H. W. Yang, and M. S. Hwang, "An enhanced secure smart card-based password authentication scheme," *International Journal of Network Security*, vol. 22, no. 2, pp. 358–363, 2020.

[18] S. Salinas and *et al.*, "Efficient secure outsourcing of large-scale sparse linear systems of equations," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 26–39, 2018.

[19] S. Salinas, C. Luo, X. Chen, and P. Li, "Efficient secure outsourcing of large-scale linear systems of equations," in *Proceedings of 2015 IEEE Conference on Computer Communications, INFOCOM 2015*, pp. 1035–1043, Hong Kong, Apr. 2015.

[20] W. L. Tai, Y. F. Chang, and W. H. Huang, "Security analyses of a data collaboration scheme with hierarchical attribute-based encryption in cloud computing," *International Journal of Network Security*, vol. 22, no. 2, pp. 212–217, 2020.

[21] C. Wang and *et al.*, "Harnessing the cloud for securely outsourcing large-scale systems of linear equations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1172–1181, 2013.

[22] Y. L. Wang, J. J. Shen, and M. S. Hwang, "A survey of reversible data hiding for VQ-compressed images," *International Journal of Network Security*, vol. 20, no. 1, pp. 1–8, 2018.

[23] H. Zhang and *et al.*, "Practical and secure outsourcing algorithms for solving quadratic congruences in internet of things," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2968–2981, 2020.

Biography

Lihua Liu, associate professor with Department of Mathematics at Shanghai Maritime University, received her PhD degree in applied mathematics from Shanghai Jiao

Tong University. Her research interests include combinatorics and cryptography.

Yujie Li is currently pursuing her master degree from Department of Mathematics at Shanghai Maritime University. Her research interests include information theory and applied mathematics.