A Heterogeneous Abstract Machine for Encrypted and Unencrypted Computation

Vaishnavi Moharir, Prof. Nitin Janwe

Department of Computer Science and Engineering, Gondwana University

ABSTRACT

The fast development and expanded notoriety of distributed computing accompanies no lack of protection worries about outsourcing calculation to semi-trusted gatherings. Utilizing the intensity of encryption, in this paper we present Cryptoleq: a dynamic machine in view of the idea of One Instruction Set Computer, fit for performing universally useful calculation on encoded programs. The program operands are ensured utilizing the Paillier somewhat homomorphic cryptosystem, which underpins expansion on the scrambled area. Full homomorphism over expansion and increase, which is important for empowering universally useful calculation, is accomplished by creating a heuristically muddled programming re-encryption module composed utilizing Cryptoleq guidelines and mixed into the executing program. Cryptoleq is varied, authorizing coming together scrambled and original recommendation operands in a similar program memory space. Programming with Cryptoleq is encouraged utilizing an upgraded low level computing construct that permits improvement of any propelled calculation on scrambled datasets. In our assessment, we analyze Cryptoleq's execution against a prevalent completely homomorphic encryption library, and show accuracy utilizing a regular Private Information Retrieval issue.

Keyword *:- Cloud*

1. INTRODUCTION

CONTEMPORARY computing paradigms, such as cloud and pervasive computing, have become increasingly popular as they allow outsourcing computation to a typically more powerful or dedicated set of machines. From Bitcoin mining [1] and Mersenne primes search [2], to commercial cloud services offered by major industry companies, outsourced computation requires code execution in a remote machine. One fundamental concern with such paradigms, however, is the privacy of the outsourced data [3]. In addition to the legitimate third party that performs the outsourced computation, additional concerns arise in light of side channel attacks [4] or even hardware Trojans [5]–[7].

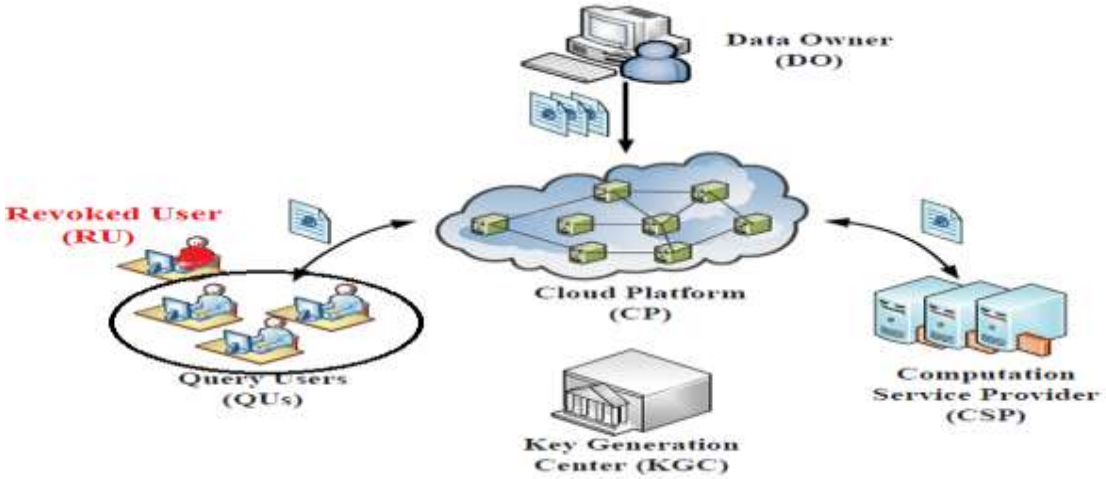
Fortunately, cryptographic primitives such as homomorphic encryption can be leveraged to address those privacy concerns, and eventually return control of the data back to the legitimate information owner [8], [9]. The academic interest in fully homomorphic encryption (FHE) applications has increased accordingly. From secure cloud computation [14] and verifiable computation [15], to multiparty computation [16] and message authenticators [17]. In addition, partial homomorphic encryption (PHE) has recently been leveraged for verifiable computation [18].

2.PROPOSED METHOD

The general approach is to encode data by the data owner (DO) before outsourcing; the approved query users (QUs) play out a perplexing arrangement of encryption and decoding operations amid query execution.

The client will scramble the lists with an indistinguishable key from the one that the data owner encodes and unscrambles the outsourced database.

It has broad applications in area based administrations, order and grouping et cetera. With the guarantee of secrecy and security, huge data are progressively outsourced to cloud in the scrambled shape for getting a charge out of the upsides of distributed computing (e.g., lessen capacity and query preparing costs). In any case, earlier works have all accepted that the query users (QUs) are completely trusted and know the key of the data owner (DO), which is utilized to scramble and decode outsourced data. The suspicions are impossible as a rule, since numerous users are neither trusted nor knowing the key.



3.Algorithm Used

Load Balancing Algorithm

1.Check the how much machines are available in the System

2. Upload File

3.Check load of every machine

4. If Load is less than particular threshold then

5. Check Memory Status of that machine.

6.If space available in that machine then upload file to that machine

7.else repeat from step 4

8.end

4.CONCLUSION

In this paper, we have displayed another computational model in view of the idea of single guideline design, ready to execute programs whose direction operands have been encoded utilizing Paillier PHE plot. General calculation is accomplished by presenting a product work, which adds increase to the theoretical machine's local option and subtraction tasks. This capacity is communicated utilizing the main accessible guideline. We have additionally built up an upgraded low level computing construct to encourage the advancement of complex projects, notwithstanding a compiler and an emulator. We assessed this system and our exploratory outcomes demonstrate that Cryptoleq acquires down to earth overhead when utilized with normal scope of substantial numbers.

Cryptoleq considers a few future changes concerning execution and security. The previous can be enhanced through the presentation of high-radix portrayals (e.g. Montgomery), and progressed runtime procedures, (for example, programmed identification of open qualities to supplant homomorphic duplication with plaintext expansion). Correspondingly, parallel obscurity is additionally a vigorously inquired about subject and future work will investigate the use of such procedures to Cryptoleq pairs to upgrade the muddling offered by our system.

5.REFERENCES

[1] M. B. Taylor, “Bitcoin and the age of bespoke silicon,” in International Conference on Compilers, Architectures and Synthesis for Embedded Systems. IEEE Press, 2013, p. 16.

[2] G. Woltman and S. Kurowski, “The great internet mersenne prime search,” [Online]. Available: http://www.mersenne.org, 2004.

[3] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, “Controlling data in the cloud: outsourcing computation without outsourcing control,” in Cloud Computing Security Workshop. ACM, 2009, pp. 85–90.

[4] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," in Computer and Communications Security (CCS), 2012, pp. 305–316.
[5] N. G. Tsoutsos, C. Konstantinou, and M. Maniatakos, "Advanced techniques for designing stealthy hardware trojans," in Design Automation Conference (DAC), 2014, pp. 1–4.
[6] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant–level hardware trojans," in Cryptographic Hardware and Embedded Systems Workshop, 2013, pp. 197–214.
[7] N. G. Tsoutsos and M. Maniatakos, "Fabrication attacks: Zero-overhead malicious modifications enabling modern microprocessor privilege escalation," IEEE Transactions on Emerging Topics in Computing, vol. 2,
no. 1, pp. 81–93, 2014.
[8] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in Advances in Cryptology– CRYPTO 2010. Springer, 2010, pp. 483–501.
[9] N. G. Tsoutsos and M. Maniatakos, "The HEROIC Framework: Encrypted Computation Without Shared Keys," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 6, pp. 875–888, 2015.
[10] C. Gentry, "Fully homomorphic encryption using ideal lattices," in ACM Symposium on Theory of Computing, 2009, pp. 169–178.
[11] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in Advances in Cryptology– EUROCRYPT 2010. Springer, 2010, pp. 24–43.
[12] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.
[13] N. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," Cryptology ePrint Archive, Report 2009/571, 2009, http://eprint.iacr.org/.
[14] M. Brenner, J. Wiebelitz, G. Von Voigt, and M. Smith, "Secret program execution in the cloud applying homomorphic encryption," in Digital Ecosystems and Technologies Conference (DEST), 2011, pp. 114–119.
[15] D. Fiore, R. Gennaro, and V. Pastro, "Efficiently verifiable computation on encrypted data," in Computer and Communications Security (CCS). ACM, 2014, pp. 844–855.
[16] A. L´opez-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in ACM Symposium on Theory of Computing. ACM, 2012, pp. 1219– 1234.
[17] R. Gennaro and D.Wichs, "Fully homomorphic message authenticators," in Advances in Cryptology-ASIACRYPT 2013. Springer, 2013, pp. 301– 320.
[18] Y. Zhang, C. Papamanthou, and J. Katz, "Alitheia: Towards practical verifiable graph processing," in Computer and Communications Security (CCS). ACM, 2014, pp. 856–867.
[19] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in Cloud Computing Security Workshop. ACM, 2011, pp. 113–124.
[20] B. Schneier, "Homomorphic encryption breakthrough," [Online]. Available: http://www.schneier.com/blog/archives/2009/07/ homomorphic enc.html, 2009, (Accessed: 11/13/15).