A PARTIAL HOMOMORPHIC ENCRYPTION TECHNIQUE FOR ENCRYPTED AND UNENCRYPTED COMPUTATION

N KEERTHANA, S KEERTHANA, S KRISHNAVENI, J JOSEPHA MENANDAS

Student , CSE department , Panimalar Engineering College, Tamilnadu, India

Student , CSE department , Panimalar Engineering College, Tamilnadu, India

Student , CSE department , Panimalar Engineering College, Tamilnadu, India

Student , CSE department , Panimalar Engineering College, Tamilnadu, India

ABSTRACT

The rapid expansion and increased popularity of cloud computing comes with no shortage of privacy concerns about outsourcing computation to semi-trusted parties. Leveraging the power of encryption, in this paper, we introduce Cryptoleq: an abstract machine based on the concept of one instruction set computer, capable of performing general-purpose computation on encrypted programs. The program operands are protected using the Paillier partially homomorphic cryptosystem, which supports addition on the encrypted domain. Full homomorphism over addition and multiplication, which is necessary for enabling general-purpose computation, is achieved by inventing a heuristically obfuscated software re-encryption module written using Cryptoleq instructions and blended into the executing program. Cryptoleq is heterogeneous, allowing mixing encrypted and unencrypted instruction operands in the same program memory space. Programming with Cryptoleq is facilitated using an enhanced assembly language that allows the development of any advanced algorithm on encrypted data sets.
In our evaluation, we compare Cryptoleq's performance against a popular fully homomorphic encryption library, and demonstrate correctness using a typical private information retrieval problem.

KEYWORD:- *crytoleq, Paillier part homomorphic cryptosystem,fully homorphy,obfuscated software system re-encryption module*

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 and Mersenne primes search , to commercial cloud services offered by major industry companies, outsourced computation requires code execution in a remote machine 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 . As soon as fully homomorphic encryption (FHE) became theoretically possible , the academic interest in FHE applications has increased accordingly. In addition, partial homomorphic encryption (PHE) has recently been leveraged for verifiable computation . Despite the wide range of applications that can benefit from FHE schemes, their efficiency has been a concern, and their practicality has been questioned. More practical implementations of FHE, such as , have already been instantiated in the HElib software library, but fully homomorphic operations could still have overheads in the order of seconds . In addition, HElib only recently has evolved to support bootstrapping and rencryption , PHE schemes are more practical than their FHE counterparts, and what the former lack in range of supported operations, they gain in efficiency. Indeed, PHE schemes typically require straightforward operations on ciphertexts, such as modular multiplication, which can be implemented very efficiently. Our observation is that PHE could be sufficient for practical applications of outsourced computation, where the applicable threat model can afford obfuscation as an adequate mitigation control to protect the privacy of processed data. Of course, PHE is less powerful than FHE in terms of computational completeness, and endto- end encryption is traded for performance in computations.

2.RELATED WORK

Richard Chow in 2009 describe how the combination of existing research thrusts has the potential to alleviate many of the concerns impeding adoption. It is a the seamless extension of control from the enterprise into the cloud through the powerful combination of high-assurance remote server integrity, and cryptographic protocols It Supports computation on ciphertext . Kai-Min Chung in 2010 a delegator outsources the computation of a function , dynamically chosen inputs to a worker in such a way that it is infeasible for the worker to make the delegator accept a result It is a "pipelined" implementation that avoids the latency of, while maintaining soundness even if accept/reject decisions are revealed . Long Chen in 2010, describe a public key encryption scheme using ideal lattices that is almost bootstrappable. A scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. It reduce the depth of the decryption circuit, and thereby obtain a bootstrappable encryption scheme It doesnot have circuit privacy .Marten van Dijk,2010 We describe a very simple "somewhat homomorphic" encryption scheme using elementary modular arithmetic, and use Gentry's techniques to convert it into a fully homomorphic scheme It uses addition and multiplication over the integers rather than working with ideal lattices over a polynomial ring. The main approach used is the conceptual simplicity. It improves efficiency .Rotter T, Kinsman L, James EL,2010 We propose to link evidence to practice and optimise clinical outcomes while maximising clinical efficiency.It assess the effect of clinical pathways on professional practice, patient outcomes, length of stay and hospital costs. Clinical pathways are document-based tools that provide a link between the best available evidence and clinical practice. They provide recommendations, processes and time-frames for the management of specific medical conditions or interventions It is used as reduction in in-hospital complications and improved documentation associated with clinical pathways we are not able to compare complex interventions including a CPW element versus a single pathway intervention in order to detect factors associated with effective pathway standardization. charges are very difficult to interpret inComparison with hospital costs and can be very misleading. Kristin Lauter,2011 It is used to compute functions of practical interest on encrypted data. It produces short ciphertexts, and its security is based on the ring learning with errors (Ring LWE) problem They are the building blocks for the FHE schemes. It computes many additions and a small number of multiplications on ciphertexts . It is much faster, and more compact than fully homomorphic encryption schemes. They provide much better efficiency It is not semantically secure. Performance is the major disadvantage .Michael Brenner,2011 propose a secret program on an untrusted resource using fully homomorphic encrypted circuits. It solves the problems of encrypted storage access with encrypted addresses and encrypted branching. It operates on encrypted functions and encrypted data. We use a **cipherspace lock up strategy** that seals program code and data entirely in the encrypted domain, which is a closed algebraic system. It supports dynamic parameters and non-linear programs. It injects data into the encrypted environment which is sufficient to receive process data from outside the cipher-space. The issue of this approach is the termination problem.The problems are correctness and consistency of the encrypted code and data . Eran Tromer,2012 propose a new notion of secure multiparty computation aided by a computationally powerful but untrusted "cloud" server. the cloud can non-interactively perform arbitrary, computations on data belonging to arbitrary sets of users chosen on-the-fly. It is capable of operating on inputs encrypted under multiple, unrelated keys. A ciphertext resulting from a multikey evaluation can be jointly decrypted using the secret keys of all the users involved in the computation. The data are protected from snooping by the cloud. It is used to evaluate any circuit on ciphertexts that might be encrypted under different public keys. The only main problem is that it requires a CRS setup. GeorgT. Becker1,2013It Implements hardware Trojans below the gate level, and we evaluate their impact on the security of the target device . It is a new type of sub-transistor level hardware Trojan that only requires modification of the dopant masks. It does not change the logic value of any gate . It do not need any extra logic resources but require a change in the dopant polarity of a few transistors. It add overhead in terms of additional transistors and metal wires. It is very difficult to find a chip that can serve as a golden chip, which is needed by most post-manufacturing Trojan detection mechanisms.

2.1EXISTING METHODOLOGY

The existing system addresses the problem of protecting the privacy of sensitive data. It is Difficult to achieve through fully homomorphic encryption(FHE) schemes. When these data are being processed within semi-trusted containers and the computation is outsourced. The technique used in existing system is Data Encryption Standard. The Data Encryption Standard was once a predominant symmetric-key algorithm for the encryption of data. It was highly influential in the advancement of modern cryptography in the world. 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 . As soon as fully homomorphic encryption (FHE) became theoretically possible , the academic interest in FHE applications has increased accordingly.

2.2 PROBLEMS IN THE EXISTING SYSTEM

Their efficiency has been a concern, and their practicality has been questioned. Fully homomorphic operations could still have overheads in the order of seconds. HElib recently has evolved to support bootstrapping and re-encryption and applications that use this library to implement generic computer programs that process homomorphic data are yet to be seen.

2.3 PROPOSED WORK

We propose Cryptoleq, a new programming language based on a single instruction computer architecture, which processes homomorphic data natively. Cryptoleq defines a universal computer for processing encrypted and unencrypted data together within the same program memory space. The technique used is advanced encryption and multiplication algorithm. The multiplication algorithm is a software library procedure that produces an encrypted product given two encrypted factors.The requirements for this algorithm are: It should be based only on addition and subtraction It cannot use any conditional jumps on encrypted values. Protected execution using encrypted data under full encryption or heuristic obfuscation modes, depending on the need to multiply encrypted values. This effectively enables unifying re-encryption with the executing program. Multiplication over encrypted values defines Function G is a software module that performs heuristically obfuscated decryption andre-encryption. Our goal is to define the simplest mathematical function which is sufficient for designing any other complex algorithm in Cryptoleq, such as encrypted multiplication or comparison.

In the user login phase, To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password, Email id, DOB and Mobile Number into the server. Database will create the account for the entire user to maintain upload and download rate. Logging is usually used to enter a specific page. It will search the query and display the query. The next phase describes This module is used to choose the file from the local System which you want to upload. It could be text file or anything. For this we need to copy the file path from the local system and paste the content into the file. Then this has been spitted into two parts and stored into the database. The next phase describes used to encrypt the splitted files. It will be encrypted by using public key. Now files encrypted successfully. Then we can see two encrypted contents. Then these encrypted contents are stored into the database. The next phase describes to upload the encrypted files into the cloud, So when ever we want the files we can get from the cloud storage, because it is online storage.The final phase describes used to download the uploaded file ,But after downloading we can't get Original file, Because file has been encrypted. So in order to get original file we need to decrypt the file.It is used to decrypt the downloaded files by using secret key. After decrypting the file you will see original file. Finally we have got our original file and we could able to get the original data without hacking by anyone.

3.CONCLUSION

Cryptoleq allows for several future improvements with regards to performance and security. The former can be improved through the introduction of high-radix representations (e.g. Montgomery), and advanced runtime techniques (such as automatic detection of open values to replace homomorphic multiplication with plaintext addition). we have presented a new computational model based on the concept of single instruction architecture, able to execute programs whose instruction operands have been encrypted using Paillier PHE scheme. Universal computation is achieved by introducing a software function, which adds multiplication to the abstract machine's native addition and subtraction operations. This function is expressed using the only available instruction.

4. REFERENCES

[1]M. B. Taylor, "Bitcoin and the age of bespoke silicon," in *Proc. Int. Conf. Compil., Archit. Synthesis Embedded Syst.*, 2013, p. 16.

[2] G. Woltman and S. Kurowski. (2004). *The Great Internet Mersenne Prime Search*. [Online]. Available: http://www.mersenne.org

[3] R. Chow *et al.*, "Controlling data in the cloud: Outsourcing computation without outsourcing control," in *Proc. ACM Cloud Comput. Secur. Workshop*, 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 *Proc. ACM Conf. Comput.Commun.Secur.(CCS)*, 2012, pp. 305–316.

[5] N. G. Tsoutsos, C. Konstantinou, and M. Maniatakos, "Advanced techniques for designing stealthy hardware trojans," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–4.

[6] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant–level hardware trojans," in *Proc. Cryptograph.Hardw.Embedded Syst. Workshop*, 2013, pp. 197–214.

[7] N. G. Tsoutsos and M. Maniatakos, "Fabrication attacks: Zero-overhead malicious modifications enabling modern microprocessor privilege escalation," *IEEE Trans. Emerging Topics Computing*, vol. 2, no. 1, pp. 81–93, Mar. 2014.

[8] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in Cryptology*. Heidelberg, Germany: Springer, 2010, pp. 483–501.

[9] N. G. Tsoutsos and M. Maniatakos, "The HEROIC framework: Encrypted computation without shared keys," *IEEE Trans. Comput.- Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 875–888, Jun. 2015.

[10] C. Gentry, "Fully homomorphicncryption using ideal lattices," in *Proc. ACM Symp. Theory Comput.*, 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*. Heidelberg, Germany: Springer, 2010, pp. 24–43.

[12] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2009.

[13] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Public Key Cryptography (PKC)*. Heidelberg, Germany: Springer, 2010, pp. 420–443.

[14] M. Brenner, J. Wiebelitz, G. von Voigt, and M. Smith, "Secret programexecution in the cloud applying homomorphic encryption," in*Proc. IEEE Int. Conf. Digit. Ecosyst. Technol. (DEST)*, May/Jun. 2011,pp. 114–119.

[15] D. Fiore, R. Gennaro, and V. Pastro, "Efficiently verifiable computationon encrypted data," in *Proc. ACM Comput. Commun.Secur.(CCS)*,2014, pp. 844–855.

[16] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multipartycomputation on the cloud via multikeyfully homomorphic encryption,"in *Proc. ACM Symp. Theory Comput.*, 2012, pp. 1219–1234.

[17] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators,"in *Advances in Cryptology*. Heidelberg, Germany: Springer, 2013,pp. 301–320.

[18] Y. Zhang, C. Papamanthou, and J. Katz, "ALITHEIA: Towards practicalverifiable graph processing," in *Proc. ACM Comput. Commun.Secur.(CCS)*, 2014, pp. 856–867.

[19] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphicencryption be practical?" in *Proc. ACM Cloud Comput.Secur. Workshop*,2011, pp. 113–124.

[20] B. Schneier. (2009). *Homomorphic Encryption Breakthrough*,accessed on Nov. 13, 15. [Online]. Available: http://www.schneier.com/blog/archives/2009/07/homomorphic_enc.html

[21] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fullyhomomorphic encryption without bootstrapping," in *Proc. Innov. Theor.Comput. Sci. Conf.*, 2012, pp. 309–325.

[22] S. Halevi and V. Shoup, "Bootstrapping for HElib," in *Advances inCryptology*. Heidelberg, Germany: Springer, 2015, pp. 641–670.

[23] S. Halevi and V. Shoup.*HElib: Design and Implementation ofa Homomorphic-Encryption Library*, accessed on Nov. 13, 2015.https://github.com/shaih/HElib