Cryptoleq’s: A Heterogeneous Abstract Machine for Encrypted and Unencrypted Computation

R.Parijatham  S.Renganayagi  R.Suganya  S.Susanna Grace
(CSE),KRCT       (CSE),KRCT       (CSE),KRCT     (CSE),KRCT
Trichy,India     Trichy,India     Trichy,India   Trichy,India

ABSTRACT:

Abstraction machine is a theoretical model of computer hardware or software system used in automatically. Then, abstract machine are used by heterogeneous for machine for memory process, ip address etc. The user proposed by encryption and decryption of same memory location. The decryption built on same memory on one set instruction computer based on multiplication process for encryption in programming languages. The encryption approach in function G generates new probabilistic cipher texts on every invocation; no plaintext information about the inputs of function G is leaked to its encrypted data. The memory process id designed by sectors of segments. In each sectors contain number of segment using availability of encrypted data with reduce waiting time machine. Then abstract machine send encrypted data to send sources node. The sources node select destination and send encrypted data. The encrypted key generated based on creation node process. Finally, the destination get original key and decrypt encrypted file and get original files.

INDEXTERM—
Abstract machine-Heterogeneous, Encryption, Decryption, Pailler formation, Partial Homomorphism encryption, Multiplication

1.INTRODUCTION

Heterogeneous abstract machine based on one of virtual machine for encrypted data. Leveraging the power of encryption, in this paper, we introduce Cryptoleq’s: 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 homomorphism 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’s instructions and blended into the executing program. cryptoleq’s heterogeneous allowing mixing encrypted and unencrypted instruction operands in the same program memory space. The sources can create abstract machine. Browse text files of data to send abstract machine. The generation of machine to encrypted data using one set instruction implement multiplication. The memory consider number of sectors and each sectors contain segment of spaces. When this process is executed by availability of memory process. The text file automatically encrypts and send to sources. Sources select destination to transmit encrypts data. Destination access key with get original files.
2. EXISTING SYSTEM

- A block cipher accelerator decrypts all oblivious RAM read operations before instructions and data move into the processor caches, while data evicted from these caches are re-encrypted before oblivious memory storage.
- The processor contains a shared block cipher key and the chip itself is considered tamper-proof.
- **Aegis** propose a secure processor, which supports integrity attestation of executed software similar to Trusted Platform Modules (tamper-evident execution), and provides privacy protection for off-chip memory using non-malleable symmetric encryption (tamper-resistant execution).
- The Aegis chip contains a permanent private key that is necessary to decrypt other symmetric keys used to protect data and instructions within program binaries, to provide privacy and integrity protection from potentially malicious processes running on the same system with elevated privileges.
- In this case, a memory encryption engine protects all traffic between the processor and the system main memory. In the same direction, the hardware enforced isolation discussed in provides integrity protection even in case the operating system kernel is under attack, using a cryptographic coprocessor provisioned with secret keys during fabrication.

3. PROPOSED SYSTEM

- Cryptoleq’s, a new programming language based on a single instruction computer architecture, which processes homomorphism data natively.
- Cryptoleq’s defines a universal computer for processing encrypted and unencrypted data together within the same program memory space.
- Data encryptions are generated using Paillier PHE, and only homomorphism addition is natively supported. Universal computation, however, requires support for both addition and multiplication, and in this work, Simulate multiplication with heuristically obfuscated reencryption implemented using Cryptoleq’s instructions.

➤ **Design and implementation of Cryptoleq’s:**

- Cryptoleq’s supports programs written without privacy protections, as well as protected execution using encrypted data under full encryption or heuristic obfuscation modes, depending on the need to multiply encrypted values.

➤ **A practical frame work for Cryptoleq’s:**

- With extended assembly language, compiler and emulator for executing cryptoleq’s program on different platforms. Cryptoleq’s is a heterogeneous, allowing mixing encrypted and unencrypted...
instruction operands in the same program memory space.

- Programming with cryptoleq’s is facilitated using an enhanced assembly language that allows the development.

4.ARCHITETURE DIAGRAM

1. Design of abstraction machine encryption files
2. Design of memory using segment.
3. Sources select Destination process
4. Destination Decryption of Required Process

![Architecture Diagram]

1. Design of abstraction machine encryption files

Abstract machines that model software are usually thought of as having very high-level operations. For example, an abstract machine that models a banking system can have operations like "deposit," "withdraw," "transfer," etc. set computer, capable of performing general-purpose. Computation on encrypted programs.

2. Design of memory using segment

The design of memory based on segment of sector. In the approach, the memory is organized as a collection of sectors. Each sector is a collection of continuous segments (i.e. sequences of memory cells) and all cell addresses within one sector share the same s value, while all cell addresses within one segment have sequential t values. Incrementing a t value by a unit returns the next cell address.

3. Sources select Destination

One instruction set computer (OISC) is a computer architecture which supports only one instruction and is able to perform universal computation. It operates on memory organized as a sequence of memory cells, while processor instructions and data reside in unified memory space, following the von-Neumann model. There exist three OISC categories (i) Transport Triggered Architecture Machines, (ii) Bit Manipulation Machines, and (iii) Arithmetic Based Machines.
4. Destination Decrypt of required process

Assembly language is a low-level programming language for a computer or other programmable device specific to particular computer architecture in contrast to most high-level programming languages, which are generally portable across multiple systems. Then final program are associated with heterogeneous abstract machine.

5. RELATED WORK

HT design methodology to achieve the above objective, namely DeTrust. Given an HT design, DeTrust keeps its original malicious behaviour while making the HT resistant to state-of-the-art hardware trust verification techniques by manipulating its trigger designs. To be specific, DeTrust implements stealthy implicit triggers for HTs by carefully spreading the trigger logic into multiple sequential levels and combinational logic blocks and combining the trigger logic with the normal logic, so that they are not easily differentiable from normal logic.

The construction of an access-driven side-channel attack by which a malicious virtual machine (VM) extracts fine-grained information from a victim VM running on the same physical computer. This attack is the first such attack demonstrated on a symmetric multiprocessing system virtualized using a modern VMM (Xen).

Such systems are very common today, ranging from desktops that use virtualization to sandbox application or OS compromises, to clouds that co-locate the workloads of mutually distrustful customers. Constructing such a side-channel requires overcoming challenges including core migration, numerous sources of channel noise, and the difficulty of pre-empting the victim with sufficient frequency to extract fine-grained information.

6. ALGORITHM DESCRIPTION

Homomorphic encryption schemes are cryptographic constructions which enable to securely perform operations on encrypted data without ever decrypting them. More precisely, a (group) homomorphism encryption scheme over a group $(G, *)$ satisfies that given two encryptions $c_1 = E_k(m_1)$ and $c_2 = E_k(m_2)$, where $m_1, m_2 \in G$ and $k$ is the encryption key, one can efficiently compute $E_k(m_1 * m_2)$ without decrypting $c_1$ and $c_2$. Homomorphic encryption schemes are widely used in many interesting applications, such as private information retrieval.

Application of Partial homomorphism encryption

Homomorphic encryption has many benefits and applications. One such benefit is that of enhanced privacy. Privacy is one of the goals of cryptography in general, but homomorphic encryption can provide even further privacy than typical encryption schemes. Consider applications in the banking world. Suppose that a customer of a bank has the total value of their accounts encrypted using their private key and that is what is stored on the bank’s servers.
One instruction set computer (OISC) is a computer architecture which supports only one instruction and is able to perform universal computation. It operates on memory organized as a sequence of memory cells, while processor instructions and data reside in unified memory space, following the von-Neumann model. There exist three OISC categories (i) Transport Triggered Architecture Machines, (ii) Bit Manipulation Machines, and (iii) Arithmetic Based Machines.
7. CONCLUSION

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 computations 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. We have also developed an enhanced assembly language to facilitate the development of complex programs, in addition to a compiler and an emulator.

8. FUTURE ENHANCEMENT

Cryptoleq’s 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). Similarly, binary obfuscation is also a heavily researched topic and future work will explore the application of such techniques to Cryptoleq binaries to enhance the obfuscation offered by our framework.

REFERENCE