VC3: Trustworthy Data Analytics in the Cloud using SGX

Felix Schuster∗ , Manuel Costa, Cedric Fournet, Christos Gkantsidis ´
Marcus Peinado, Gloria Mainar-Ruiz, Mark Russinovich

Microsoft Research

Outline

- Introduction
- Background
- Design Overview
- Job Deployment
- Job Execution and Verification
- Regional Self-Integrity
- Implementation
- Evaluation
- Related Work
- Conclusion

Introduction

- Cloud providers allow computers into data centers and make them available on-demand
- Users have the ability to rent out computing capacity to run large-scale distributed computations based on frameworks like MapReduce
- A major concern for users is the ability to trust the cloud provider with their code and data

Introduction (cont’d)

- Concerns:
  - Single malicious insider with admin access in the cloud can leak or manipulate sensitive user data
  - External attackers attempt to access data (e.g. exploit vulnerabilities in an OS)
  - External attackers may tamper with users’ computations
- Cloud User Expectations
  - Confidentiality and integrity for both code and data
  - Verifiability of execution of the code over data
- Multiparty computation techniques may address these demands using **Fully Homomorphic Encryption** (FHE)
  - However, FHE is not efficient for most computations

Introduction (cont'd)

- **Verifiable Confidential Cloud Computing (VC3)**
  - A system that allows users to run MapReduce computations in the cloud while keeping their code and data secret and ensuring correctness and completeness of their results
- Threat Model
  - Powerful attackers that may have the ability to control the whole cloud providers software and hardware infrastructure
- Tools Used
  - Trusted SGX processors
  - Ran an unmodified Hadoop

Introduction (cont’d)

- Challenges:
  - Partition the system into trusted and untrusted parts to minimize its TCB
  - Guarantee integrity for the whole distributed computation
  - Protect the code running in the isolated memory regions from attacks due to unsafe memory accesses

Background

- MapReduce
  - A popular programming model for processing large data sets: users write map and reduce functions, and execution of functions is automatically parallelized and distributed
- Intel SGX
  - Set of x86-64 ISA extensions
    - Sets up protected execution environments (called enclaves) without requiring trust in anything but processor and code put in the enclaves

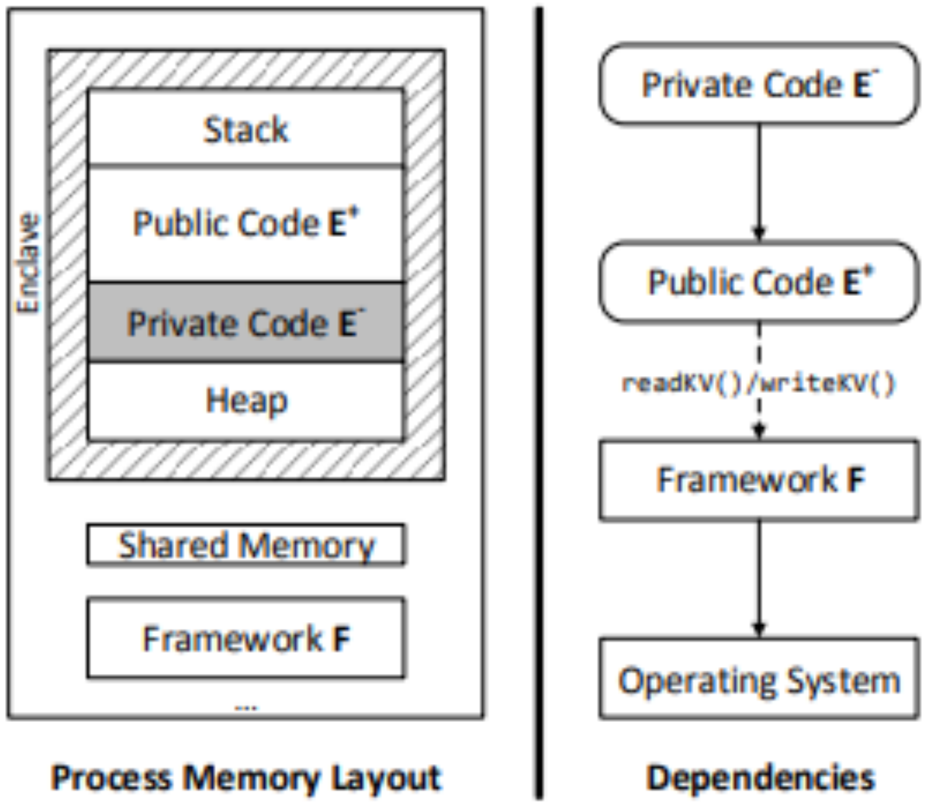
Adversary Model

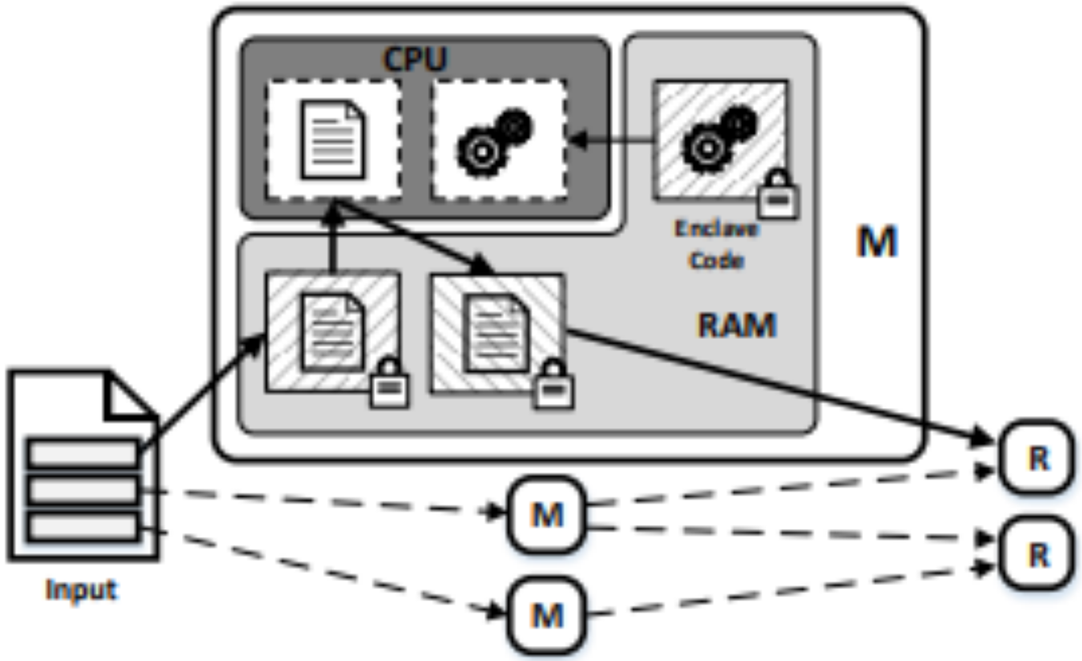
- Aware of external attackers that may try to control the entire software stack in a cloud provider’s infrastructure, including the hypervisor and OS
- Assume the attacker is unable to physically open and manipulate tat least the SGX-enabled processor packages

Design Overview

- Goal: Maintain confidentiality and integrity of code and data
- Researchers designed VC3 to achieve good performance and keep large software components out of the TCB
- VC3 allows users to implement MapReduce jobs by writing, testing, and debugging map and reduce functions
- When map and reduce functions are ready for production, users compile and encrypt the code, and obtain a private enclave *E-* code
- In the cloud, enclaves containing *E-* and *E+* are initialized and l

Design Overview





Job Deployment

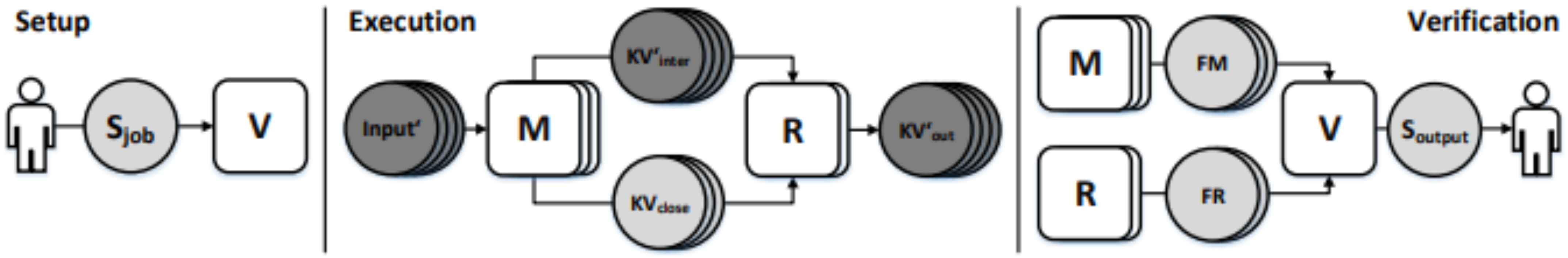
- After the deployment of a users code to the cloud, cryptographic protocols are exchanged and the actual MapReduce job execution starts
- Cloud Attestation
    - SGX remote attestation for enclaves is achieved through *quotes* issued by QE
    - Threat model excludes physical attacks, to defend against such attacks, they used an additional Cloud QE
    - Cloud QE was created by the cloud provider when a new SGX-enabled system is created

Job Deployment

- Key Exchange
  - To execute MapReduce jobs, enclaves need to get keys to decrypt the results
  - Researchers created their own key exchange protocol which is designed to implement a conventional MapReduce job that works with Hadoop

Job Execution & Verification

- Key exchanges and encryption code will help code and data be safe from attacks
- Researchers have to encrypt data in a MapReduce job and this capability needs to work within Hadoop



Region Self-Integrity

- Final aspect of design is to enforce a region of self-integrity for user code loaded into enclaves
- Establish efficient communication channels
  - Leads to a broaden attack surface on enclaves
- Two solutions:
  - Region-write-integrity
  - Region-read-write-integrity

Discussion

- Several Attack Scenarios:
  - Information Leakage
    - One basic principle of MapReduce is that key-value pairs with the same key need to be processed by the same reducer
    - A network attacker can count the number of pairs being delivered and change the pairs
  - Replay Attacks
    - Attackers can try to fully or partially replay a past MapReduce job

Implementation

- VC3 was implemented using C++ for Windows 64-bit and HDInsight distribution of Hadoop
- SGX Emulation
  - Researchers implemented VC3 in an SGX Emulator which was successful
  - As well, created their own emulator, however the emulator does not provide security guarantees

Evaluation

- Researchers chose a mix of real-world applications and benchmarks to evaluate the VC3 system
- The following table shows the applications used to evaluate VC3

Application	LLOC	Size input	Size E^- (vc3)	map tasks
UserUsage	224	41 GB	18 KB	665
IoVolumes	241	94 GB	16 KB	1530
Options	6098	1.4 MB	42 KB	96
WordCount	103	10 GB	18 KB	162
Pi	88	8.8 MB	15 KB	16
Revenue	96	70 GB	16 KB	256
KeySearch	125	1.4 MB	12 KB	96

TABLE I: Applications used to evaluate *VC3*.

Conclusion

- VC3 created as an approach for the verifiable and confidential execution of MapReduce jobs in untrusted cloud environments
- VC3 is able to be successful implemented and has strong security guarantees
- VC3 is able to achieve secure cloud computations