CS 2530 - Computer and Network Security Advanced Topics

VC3: Trustworthy Data Analytics in the Cloud Using Intel SGX

Mohammad H. Mofrad & Spencer L. Gray

University of Pittsburgh

Thursday, December 1, 2016

hasanzadeh@cs.pitt.edu & slg94@cs.pitt.edu

Preface

- Introduction
  - Article information
  - Cloud security challenges
  - Intel SGX
  - MapReduce framework
- VC3 (Verifiable Confidential Cloud Computing)
  - Threat model
  - Design overview
  - Region integrity
  - Job deployment
  - Job execution and verification
  - Results

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Article Information

Title: VC3: trustworthy data analytics in the cloud using SGX

Conference: 36^{th} IEEE Symposium on Security & Privacy

h5-index: 59

Authors: F. Schuster et al

Affiliation: Microsoft Research

Partnership: Intel Corporation

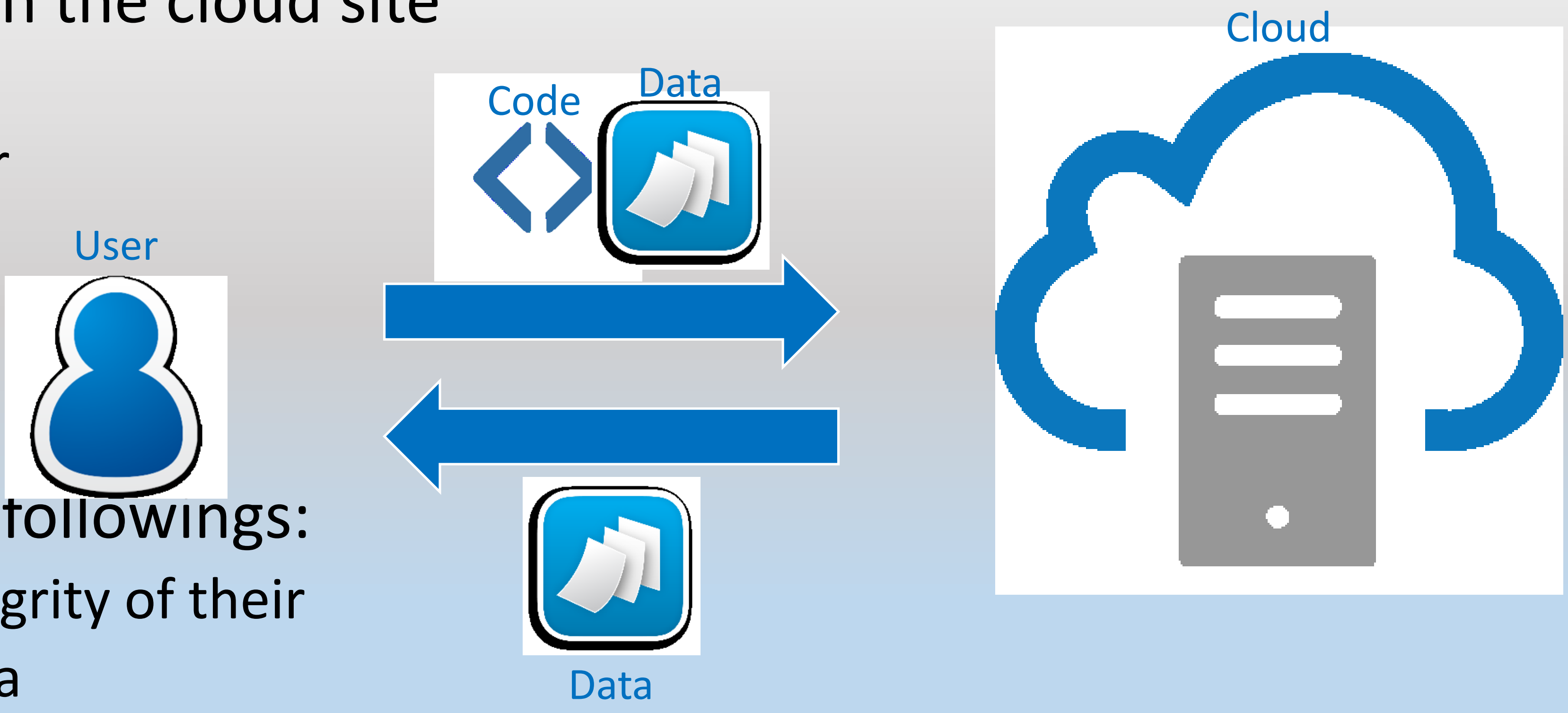
Citation: 56

Year: 2015

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Cloud Computing Security Challenges

- Information leakage in the cloud site
  - Malicious cloud
  - Malicious admin user
  - Attacks
  - Government



- Cloud users seek the followings:
  - Confidentiality + Integrity of their code & data
  - Verifiability of execution of the code over data

F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Threat Model

- Assumes a strong, but not all powerful adversary
- Adversary can:
  - Control the software stack (the hypervisor and the OS)
  - Physically access the hardware of an SGX processor
  - Read/modify any data leaving a processor
  - Read/modify/replay any network packets
  - Access any job running on the cloud simultaneously

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

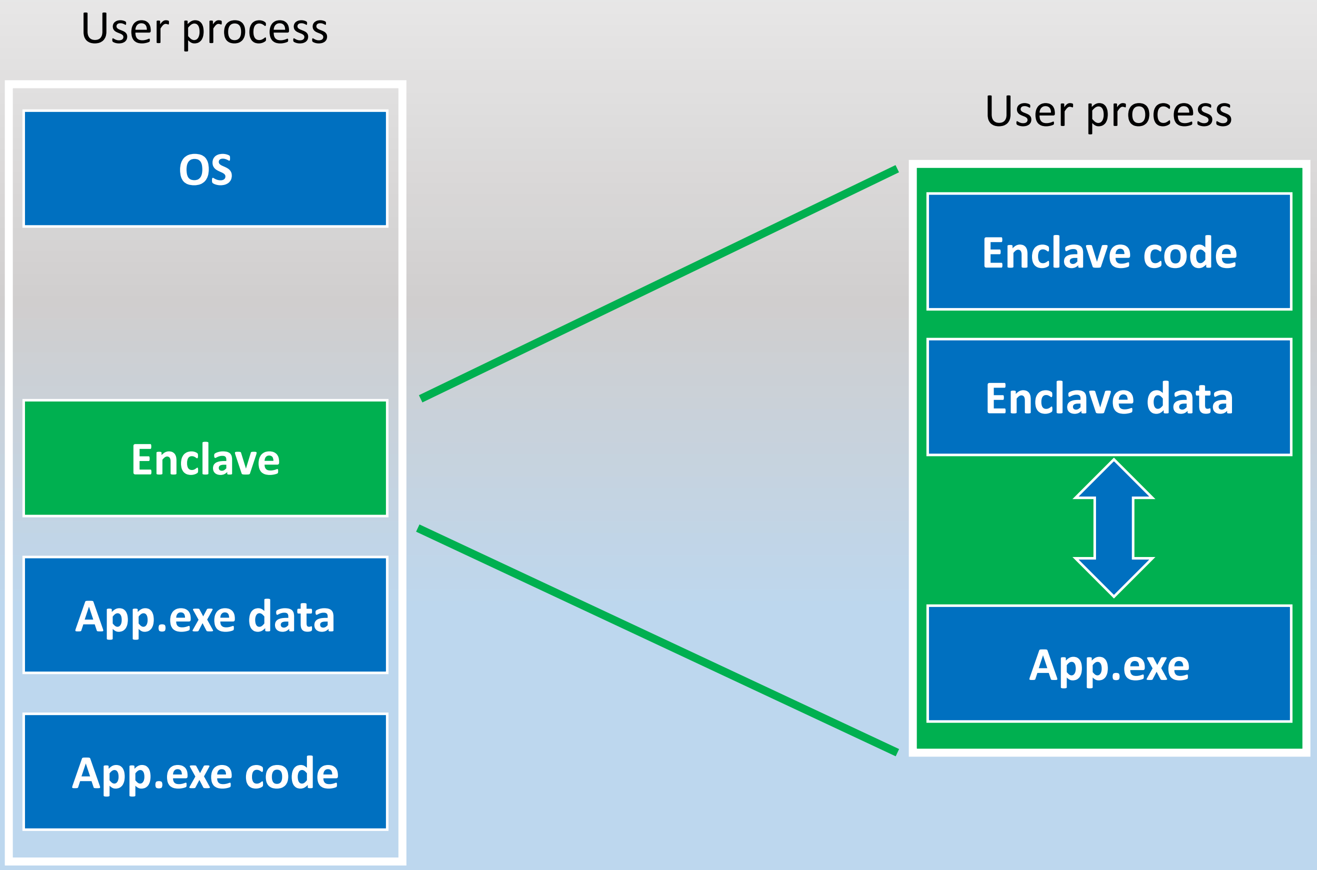
Threat Model

- There are some limitations
- Adversary can’t
  - Access the hardware of any machine at the data center
  - Denial of Service attack
  - Analyze network traffic
  - Side channel attacks
  - Fault injections
- Users are assumed to write code reasonably well
  - Assuming users can only have low level defects
  - Users will not intentionally leak information

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Intel Software Guard Extension (SGX)

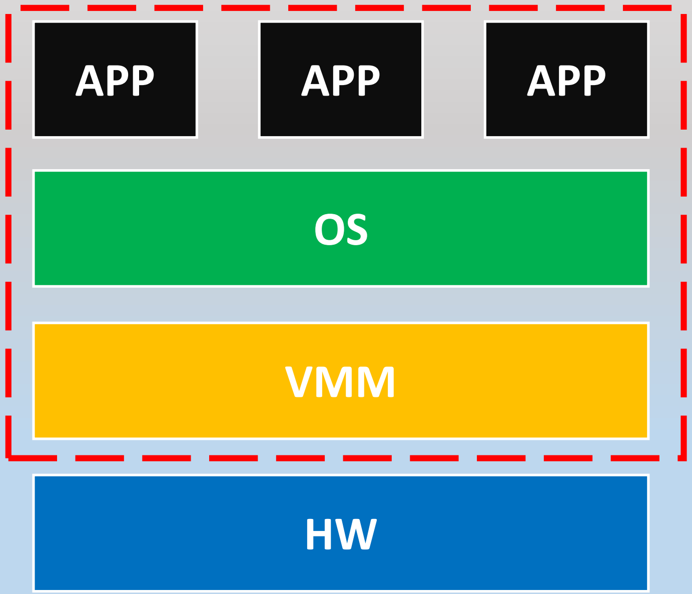
- **ENCLAVE?**
- Hardware-based protection
- User level execution
- Small TCB



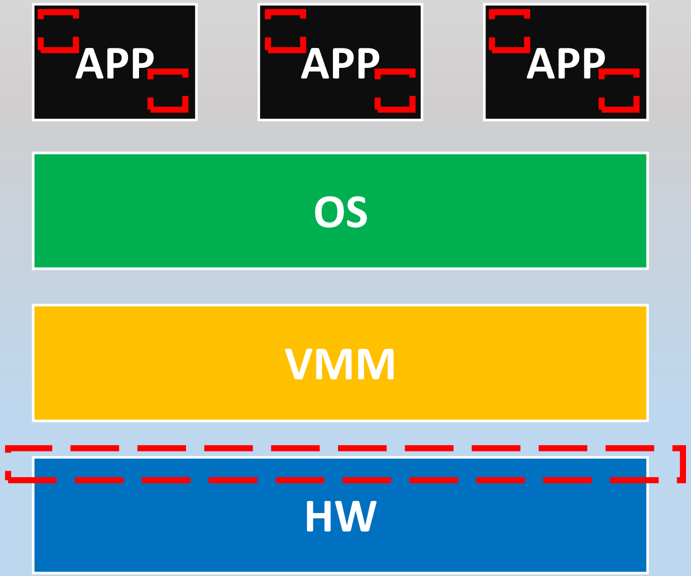
F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Reduced attack surface with Intel SGX

Attack surface of a computer system



Attack surface using Enclave

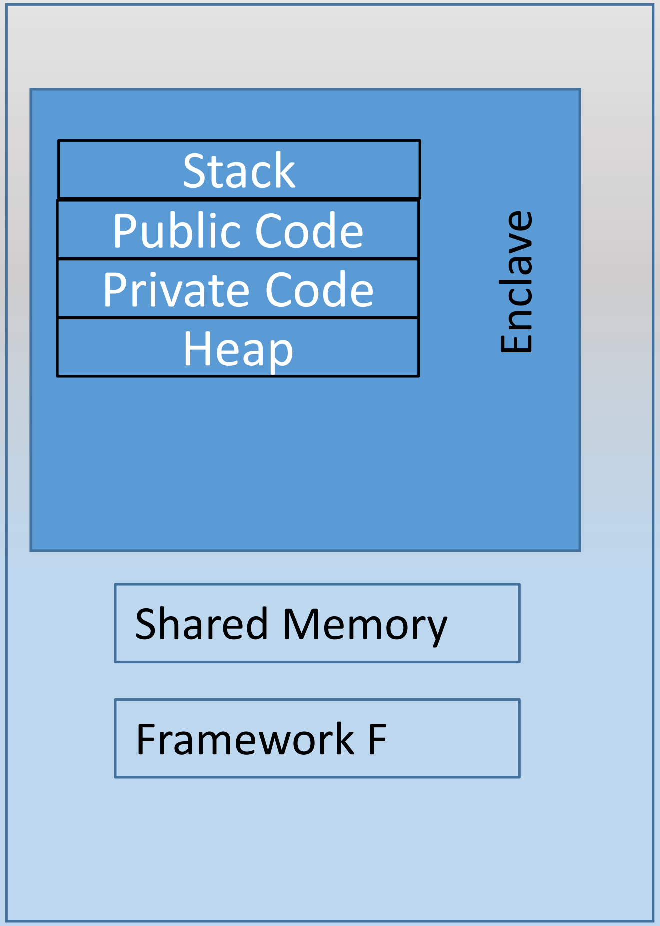


Attack surface

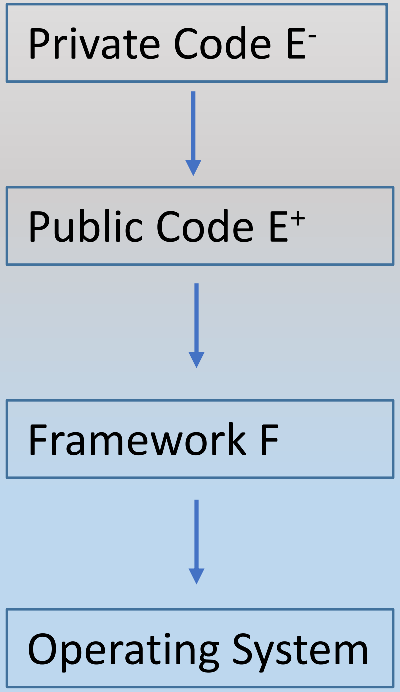
F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Design Overview

- E^- = user written map and reduce functions compiled and encrypted
  - Data is always encrypted when on cloud
  - Is only in plaintext when it is being run on a trusted processor chip
- E^+ = public code for key exchange and execution protocols
  - The operating system is not a part of E^+ to reduce trusted code base
  - The enclave code is designed to not need the OS
- F = untrusted code for worker nodes



Process Memory Layout

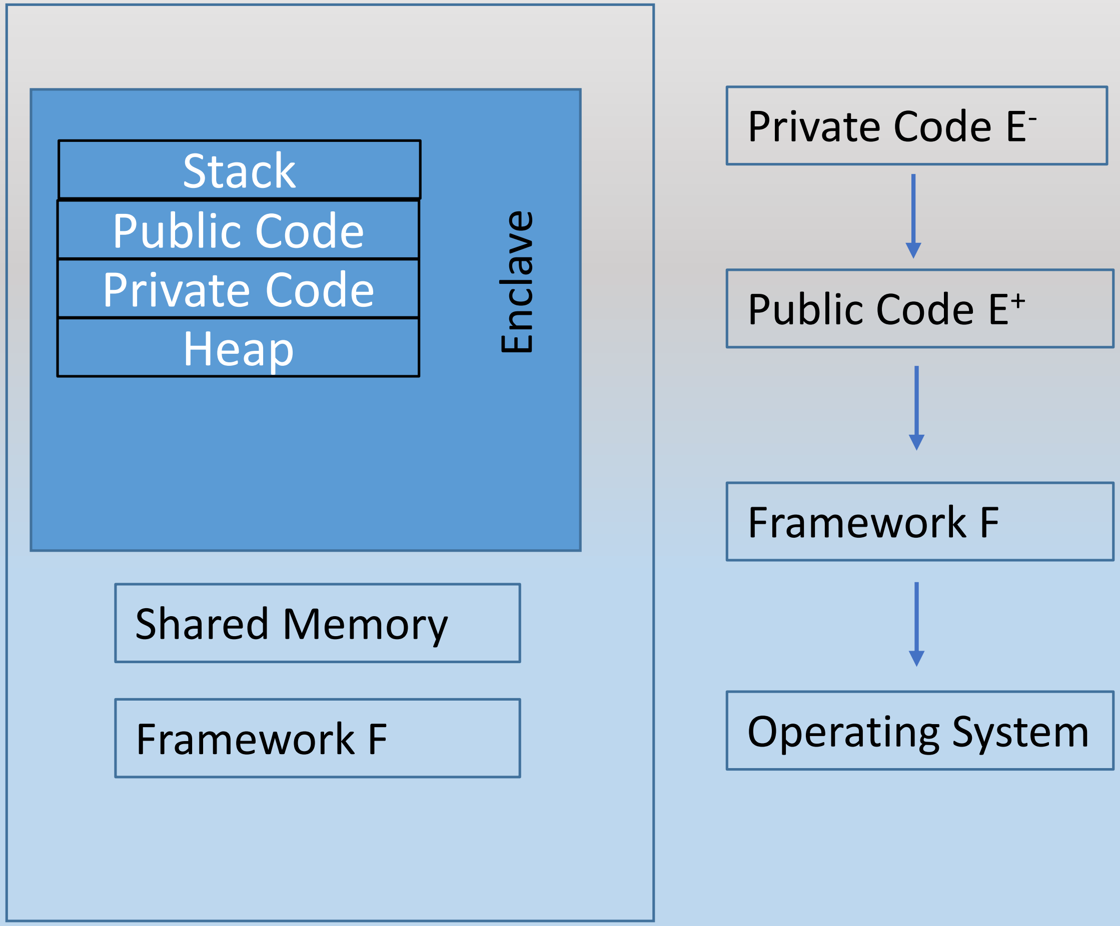


Dependencies

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Design Overview

- Enclaves contain their own stack and heap, but share memory with F
  - This allows E^+ code to interact outside of the enclave
- F helps run the enclave, but **region self-integrity** prevents it from corrupting enclave data

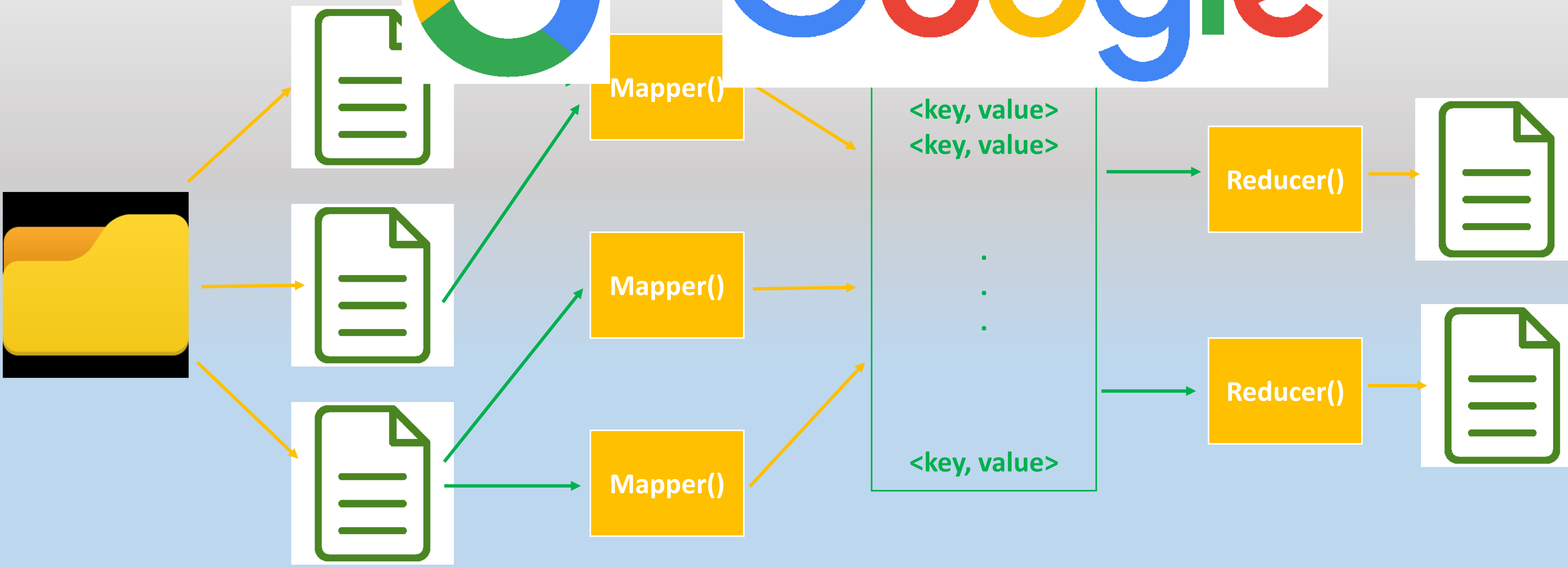


Process Memory Layout

Dependencies

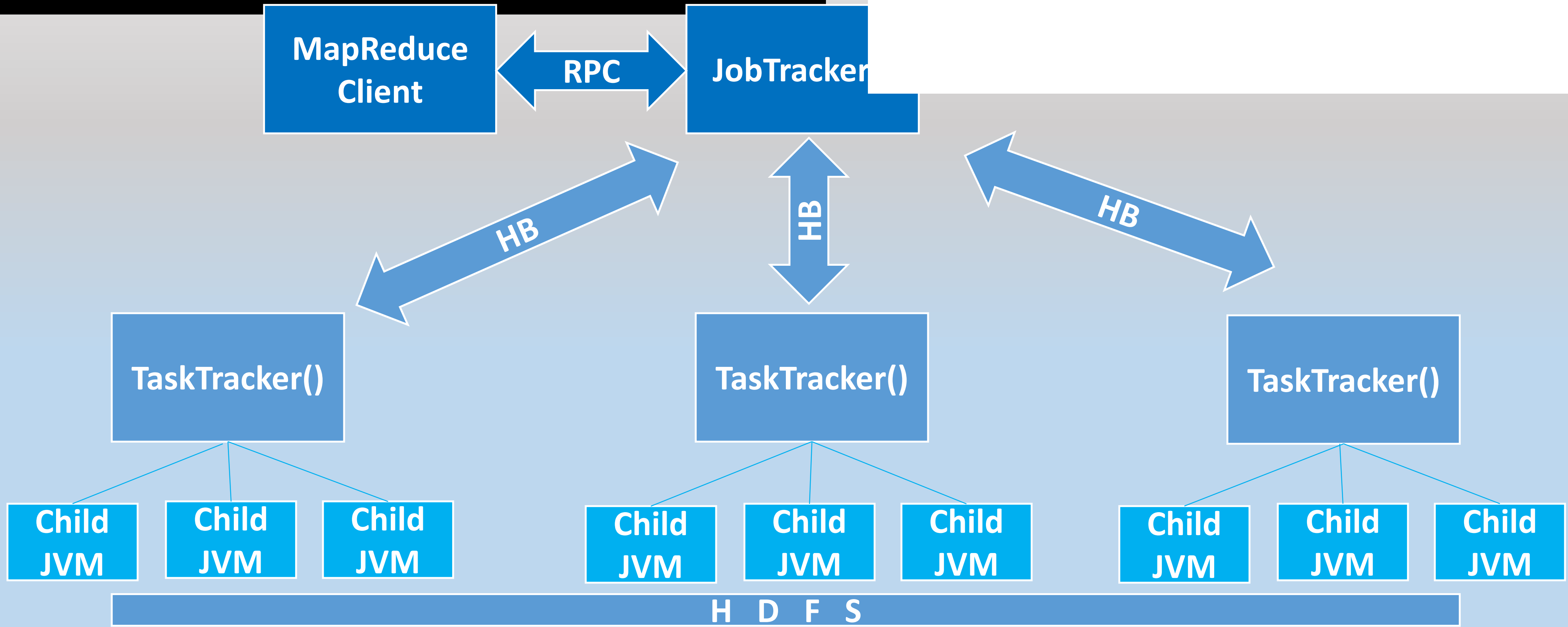
F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

MapReduce Framework



F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Hadoop MapReduce



F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Motivation Behind Region Self-Integrity

- Enclave code shares virtual address space with untrusted code
  - Allows efficient communication outside of the enclave
  - Broadens attack surface
- Enclaves could dereference a corrupted pointer from the shared memory region
- Enclave could leak data in this way

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Region Read-Write-Integrity

- Region Write-Integrity
  - Guarantees that any write that uses a pointer only writes to variables whose addresses are used in the code
  - Example: Using a pointer to write to &var_name or using an array
  - Prevents memory corruption and memory leaks
- Region Read-Write-Integrity
  - Same write guarantee as above
  - Reading a pointer only reads an address inside the enclave
  - Prevents data from being injected into the enclave

F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Enforcing the Integrity

- Both Region-Read and Region-Read-Write Integrities are enforced by a special compiler
- Code in the enclaves is written by VC3 and the user
  - We compile the code in E^- and E^+, not the cloud
- Compiler inserts code for dynamic evaluations of these integrities
  - Can’t always determine a violation statically
- When a violation is found, program is stopped (safer)
  - Could have also just masked the bits in the address
- Tradeoff between security and efficiency

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Cryptographic Assumptions

- `m | n`: Concatenation of two messages m & n
- $\mathtt{PRF}_k\mathtt{(text)}$: pseudo-random function
- `H(text)`: Collision-resistant cryptographic hash (HMAC + SHA-256)
- `EDigest(C)`: Digest of an enclave's initial content C
- `PKGen`: Creates public key p_k, s_k
- $\mathtt{PKEnc}_{pk}\mathtt{(text)}$: Encrypts *text* under p_k
- $\mathtt{Esig}_p\mathtt{(c)}$: The identity *p* jointly signs `H(text)` and `EDigest(C)`
- $\mathtt{Enc}_k\mathtt{(text,\ data)}$: Encryption of text with associated data
- $\mathtt{Dec}_k\mathtt{(cipher,\ data)}$: Decryption of cipher with associated data

F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Job Deployment

- Environment
  - Working on a unmodified version of Hadoop
  - Small TCB by Keeping Hadoop, OS, and hypervisor out of the TCB

Trusted part

1. `Map()` & `reduce()` functions
2. Encryption them
3. Bind them with some code that enables some cryptographic operations
4. Upload them on the Cloud

Untrusted part

1. In a *Worker Node*, the Cloud OS loads the code in an Enclave
2. The hosted code in the enclave will run a key exchange protocol
3. The Hadoop will run the code in the distributed manner

F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Job Deployment - Cloud attestation

- Processors
  - **Quote Enclave** for inter-platform enclave attestation
  - A symmetric key that verifies the enclave authenticity (genuine SGX processor)
- Machines
  - Cloud QE is a pair of public/private key for each
  - Public key + sealed private key in the Cloud QE
- Create an identity from Processor + Machine
  - $\texttt{Esig}_{\texttt{SGX}}\texttt{(C, text) | Esig}_{\texttt{cloud}}\texttt{(C, text)}$

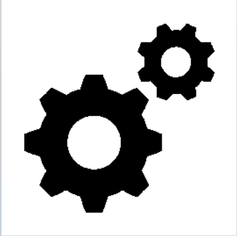
F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Job Deployment – Key Exchange

- Existing Hadoop's communication channel

user → mapper → reducer → user

- VC3's MapReduce Job (e.g. a MapTask):

`Enc`$_k$`(code, data)`  `Dec`$_k$`(code, data)`  `Enc`$_k$`(result)`

- Adding an in-band variant of key exchange
- Lightweight key exchange job before the actual job
  1. Run key exchange job ($C_{j,u}$)
  2. Create job credential (JC_{w})
  3. Run actual job (J_w)

Job Deployment – Setting up a MapReduce Job

1. User authentication pk_u
2. Invoking nodes to compute the job enclave ($C_{j,u}$)

   $C_{j,u} = E^+ \mid \mathtt{Enc}_{k\,\mathrm{code}}(E^-) \mid j \mid pk_u$
3. Each node w starts working on and create the m_w message

   $m_w = \mathtt{PKEnc}_{pku}(k_w)$

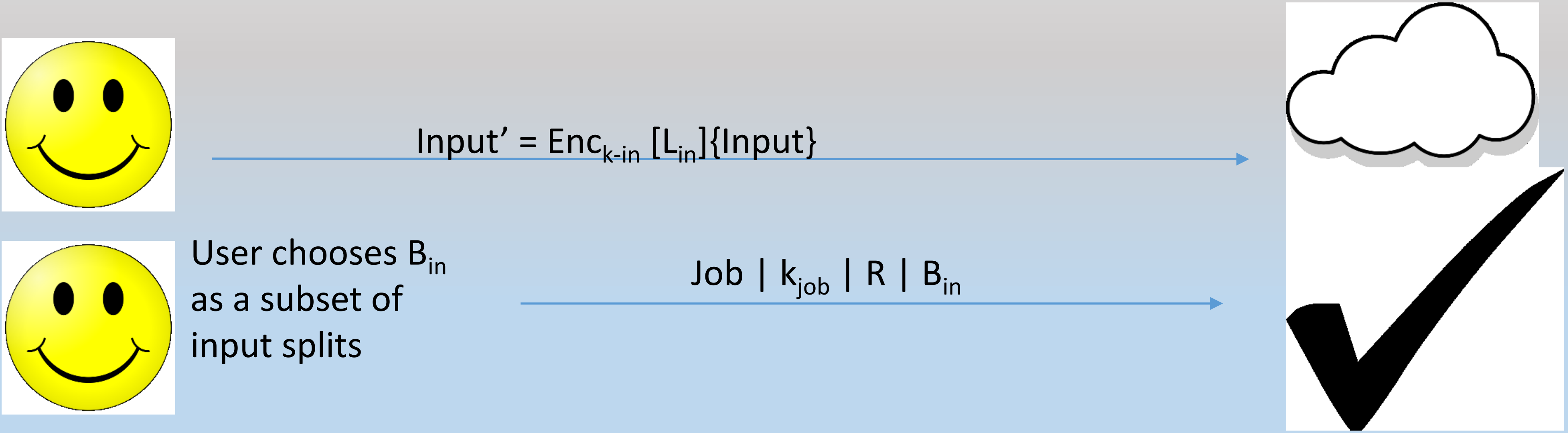
   Then requests quotes from the SGX and Cloud QE

   $p_w = m_w \mid \mathtt{ESig}_{SGX\ Cloud}(C_{j,u})\ (m_w)$
4. The task process verifies the p_w and signs the code identity $C_{j,u}$ and creates Job credential JC_w

   $JC_w = \mathtt{Enc}_{kw}(\ K_{code}, k)$ where $k = k_{job} \mid k_{in} \mid k_{inter} \mid k_{out} \mid k_{prf}$
5. Each node resumes E^+ and decrypts the JC_w

Job Execution and Verification - Set Up

- Each input split is bound to a fresh unique ID L_{in}
- R = Number of logical reducers for job
- The user encrypts each input split :



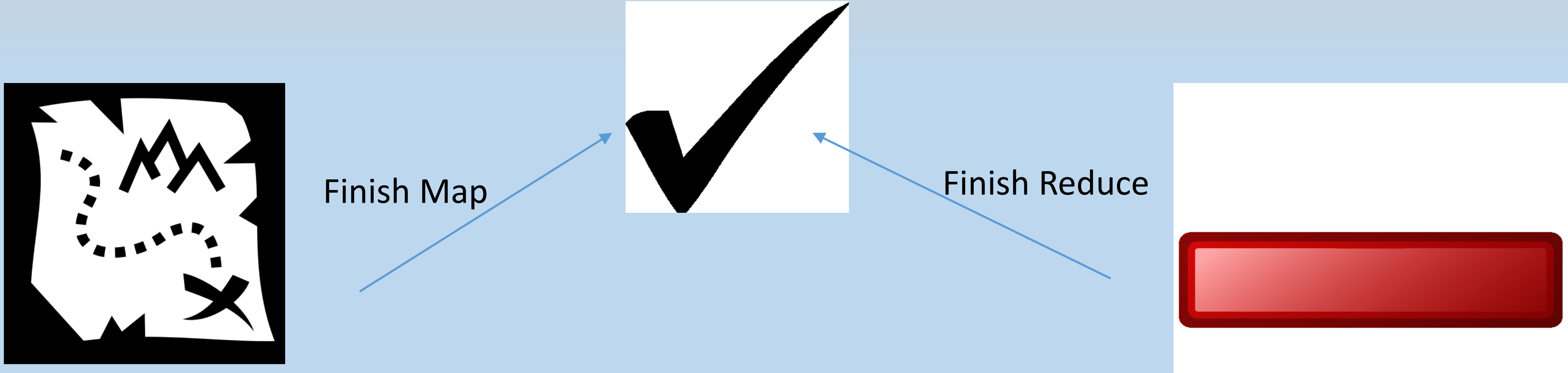
Job Execution and Verification - Mapping

- Mapper in the cloud receive splits
  - Keep track of ID’s to prevent replay attacks
- Mappers produce intermediate key-value pairs
  - Key-value pairs with identical keys must be processed by same reducer
- After processing inputs, mappers create a closing intermediate key value pair
  - Allows a reducer to exit if it receives duplicate key-value pairs or not enough key-value pairs
  - Detects if cloud services dropping jobs or attempting replay attack

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Job Execution and Verification - Verification

- Verifier receives map messages and reducer messages
- Verifies integrity of job by checking first to see if it received correct amount of each
- Verifier prevents replay attacks between different jobs
  - Each message contains a job specific ID
- Performance Cost for verification is very small



F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015

Evaluation

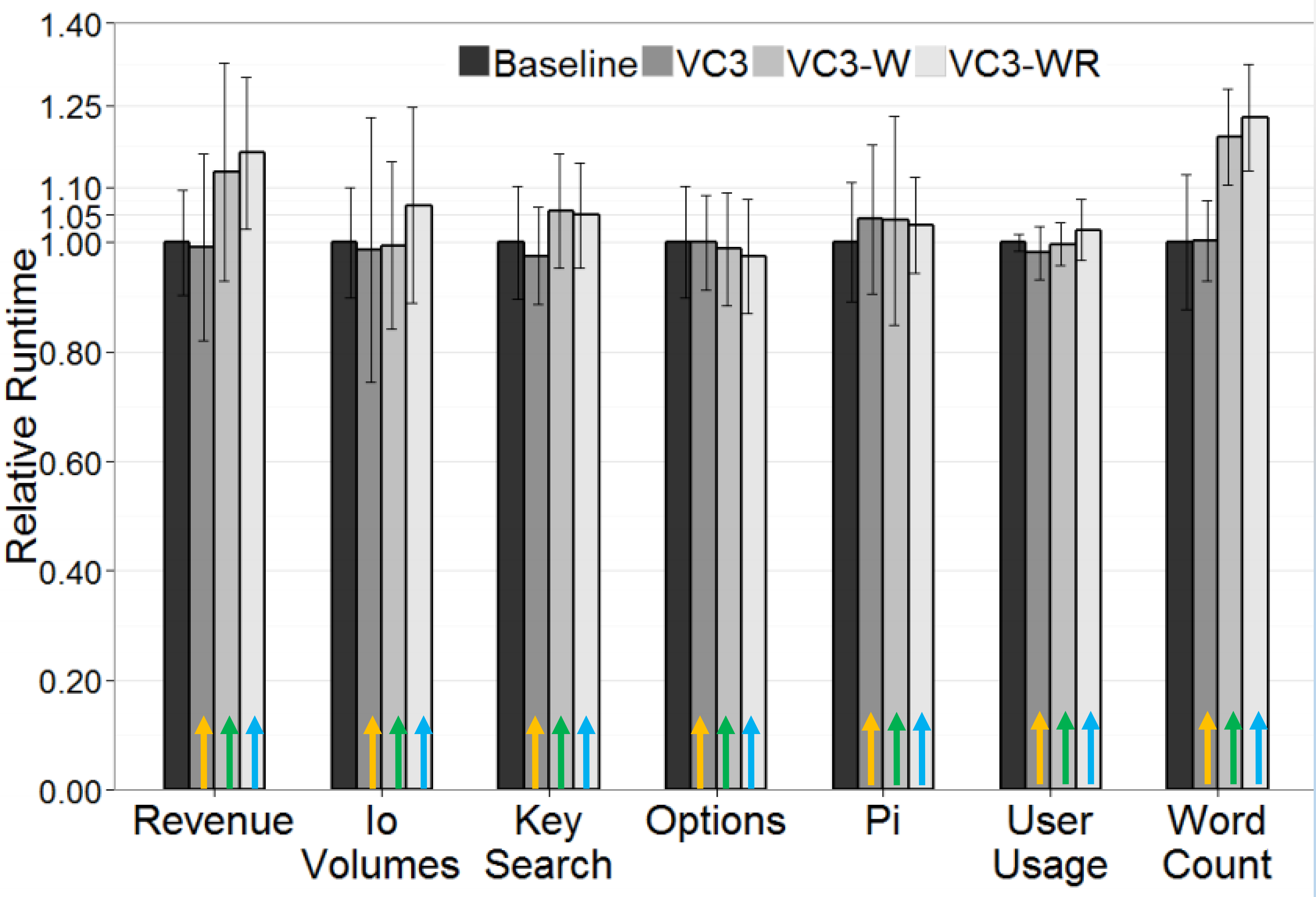
- Deploy Hadoop on a cluster of 8 machines
  - Windows server 2016 R2 64-bit
  - CPU: 2.9 GHz Core i5 Haswell family
  - HDD: 250GB SSD capability
  - Benchmarks: 7 I/O and computation intensive programs

Application	LLOC	Input size	E^- size	# map tasks
User usage	224	41 GB	18 KB	665
I/O volumes	241	94 GB	16 KB	1530
Options	6098	1.4 MB	42 KB	96
Word count	103	10 GB	18 KB	162
Pi	88	8.8 MB	15 KB	16
Revenue	96	70 GB	16 KB	256
Key search	125	1.4 MB	12 KB	96

F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Experiments

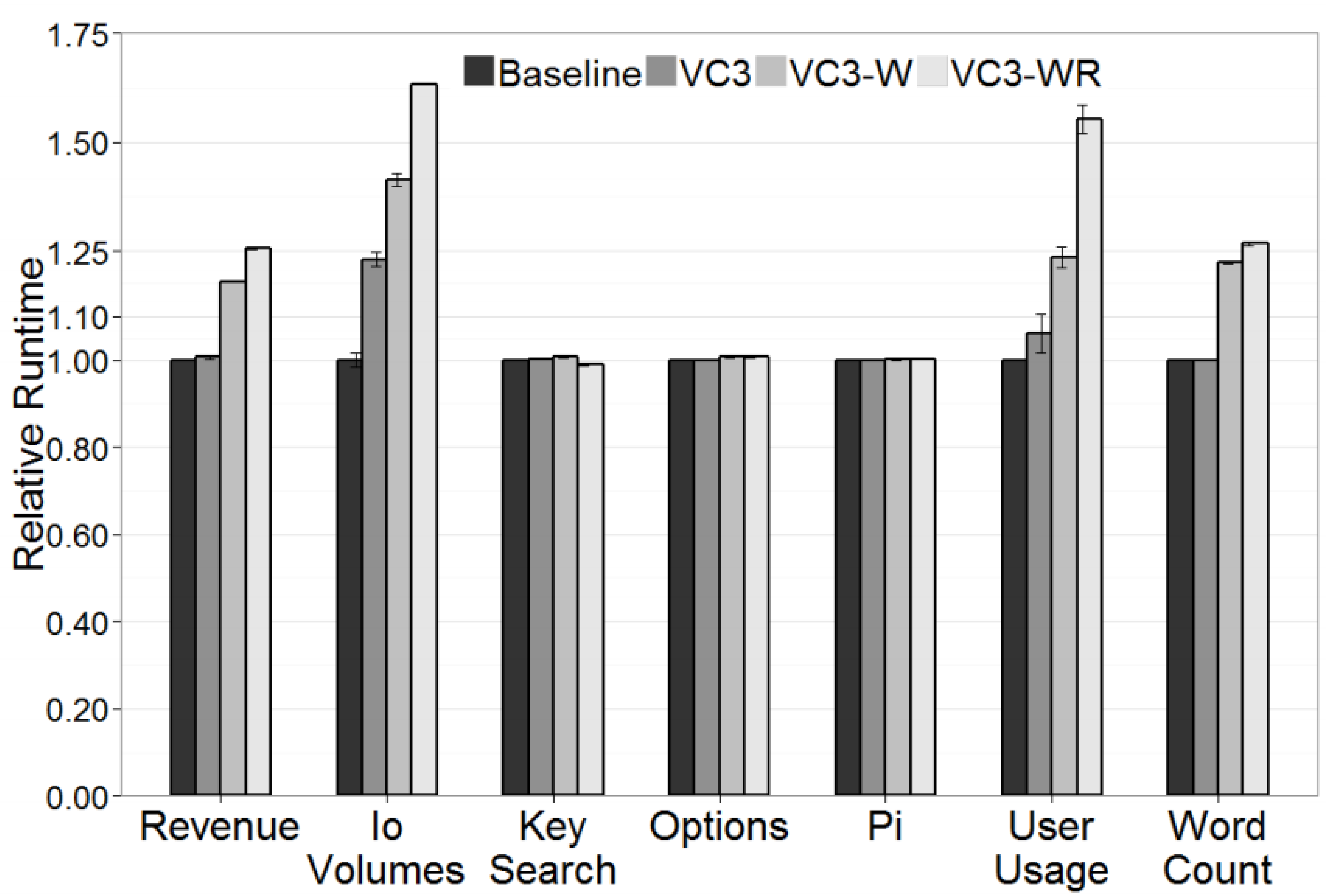
- Execution time of running MapReduce jobs in the Hadoop cluster
- `Baseline` Hadoop
- **VC3** with encrypted mapper & reducer inputs & outputs (1% avg. overhead)
- **W** (region write integrity) 4.5%
- **WR** (region read-write integrity) 8%



F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Experiments

- Execution time of running the map phase of MapReduce jobs in **isolation**
- Eliminating most of …
  - Hadoop internal I/Os
  - Enclave operations
  - Cryptographic Ops.



Discussion

- **Good design decision:**
  - Keeping TCB small
  - IND-CPA assumption
- VC3 **threat model** assumptions:
  - hardware attacks, e.g. power analysis
  - Side-channel
  - Replay attack
- **Encryption overhead**
  - cryptographic operations
  - Copying data to and from enclave
  - Enclave creation & transitions
  - Trade off between performance and security
- Simulating SGX capabilities at that time:
  - Intel SGX emulator
  - Microsoft C++ compiler

F.Schuster et al. "VC3: trustworthy data analytics in the cloud using SGX," 36th IEEE Symposium on Security & Privacy, 2015

Discussion

- Combining clear hardware security with the cloud brings **confidentiality** and **integrity** to the Cloud but …
  - **Information leakage**: Learning the intermediate <key, value> distribution

MapReduce job → (A set of map tasks) → (A set of reduce taks)

FIX: Padding, clustering, and shuffling

  - **Replay attacks**: Having $C_{j,u}$ and CJ_w and adversary can launch arbitrarily reply parts of a job that the processor participated before

This amplify other side-channel attacks against confidentiality

FIX: hardcoding the job's specification into mapper

Discussion

- **VC3** Brings data **confidentiality** and **integrity** using cryptographic operations
  - **Advantage**: Even if an adversary or a privileged Cloud user access data, they will just see encrypted data.
  - **Advantage**: The Cloud provider cannot fake the results and the customer can **verify** the results of computation
- Working on an unmodified version of **Hadoop** make it easy for research community to easily attach their framework to the existing Hadoop-based products.

Intel SGX

Microsoft Windows

Yahoo Hadoop

F.Schuster et al. “VC3: trustworthy data analytics in the cloud using SGX,” 36th IEEE Symposium on Security & Privacy, 2015