VC3: Trustworthy Data Analytics in the Cloud using SGX

Felix Schuster*, Manuel Costa, C´edric Fournet, Christos Gkantsidis Marcus Peinado, Gloria Mainar-Ruiz, Mark Russinovich
Microsoft Research

Presenter: Lun Wang
The New Role of Data
Trust the Service Provider?

• Facebook Data Privacy Scandal

• Uber Data Breach

• Equifax Data Breach

• Google Is Found to Collect Android Location Data
Cloud + TEE!
Map Reduce Model

MapReduce Word Count Process
Quote Enclave

- Quote:
  - Signature
  - SK can only be get by trusted member
  - PK is published by some corporation

- SGX Quote Enclave
  - PK published by Intel
  - “I am on a genuine SGX!”

- Cloud Quote Enclave
  - PK published by Cloud Provider
  - “I am on the cloud!”

Sometimes you will never know the value of a moment until it becomes a memory.

-Dr. Seuss
Authenticated Encryption with Associated Data

\[ \text{Enc}_k[ad]\{text\} \]

- **Confidentiality**
- **Integrity**
- **Authentication**
- **Associated Data**
Threat Model

- Powerful adversaries:
  - hardware + software
  - except processor

- Denial-of-service
- Side-channels
- Traffic-analysis
Security Guarantees

- Confidentiality
- Integrity
- Verifiability of execution
Design Overview

- Write Mapper & Reducer Function
- Compile & Encrypt (Region Self-Integrity)
- Load Codes into Workers
- Job Execution
- Key Exchange
- Job Verification
Design Overview

- Write Mapper & Reducer Function
- Compile & Encrypt (Region Self-Integrity)
- Load Codes into Workers
- Job Verification
- Job Execution
- Key Exchange
Key Exchange

\[ C_{j,u} = E^+ | \text{Enc}_{k_{code}} \{ E^- \} | j | p_{k_u}. \]
Key Exchange

\[ m_w = \text{PKEnc}_{p_{kw}} \{k_w\} \]

\[ p_w = m_w | \text{ESig}_{SGX,Cloud}[C_{j,u}]{m_w} \]
Key Exchange

Why they use so many keys?

\[ JC_w = \text{Enc}_{k_w}[\{k_{code} \mid k\} \mid k = k_{job} \mid k_{in} \mid k_{inter} \mid k_{out} \mid k_{prf}] \]
Design Overview

- Write Mapper & Reducer Function
- Compile & Encrypt (Region Self-Integrity)
- Load Codes into Workers
- Job Execution
- Key Exchange
- Job Verification
Why we still need the r in the KVclose value?

Job Execution: Mapping

\[
K'_{\text{inter}} = r \equiv \text{PRF}_{k_{\text{prf}}}(K_{\text{inter}}) \mod R
\]

\[
V'_{\text{inter}} = \text{Enc}_{k_{\text{inter}}}[j \mid \ell_m \mid r \mid i_m,r]\{\langle K_{\text{inter}} : V_{\text{inter}} \rangle\}
\]

\[
KV'_{\text{inter}} = \langle K'_{\text{inter}} : V'_{\text{inter}} \rangle
\]

\[
KV_{\text{close}} = \langle r : \text{Enc}_{k_{\text{inter}}}[j \mid \ell_m \mid r \mid i_m,r]\{\}angle
\]
Job Execution: Reducing

\[ KV'_\text{out} = \langle l_{\text{out}} : \text{Enc}_{k_{\text{out}}}[l_{\text{out}}]\{KV_{\text{out}}\} \rangle \]
Design Overview

Write Mapper & Reducer Function → Compile & Encrypt (Region Self-Integrity) → Load Codes into Workers

Load Codes into Workers → Key Exchange → Job Execution → Job Verification
Job Setup

Key for Verification Message

\[ j \mid k_{job} \mid \hat{R} \mid B_{in} \]

Job Identifier

Input Indexes

Number of Reducers

Setup

\( S_{job} \rightarrow V \)
Job Verification: Mapper

\[ FM = \text{Enc}_{k_{job}}[j \mid \ell_m \mid B_{in,m}] \]
Job Verification: Reducer

\[
FR = j \mid r \mid B_{out,r} \mid \text{Enc}_k(j \mid r \mid B_{out,r} \mid P_r, \{\}) \\
Pr \subseteq (l_m)_{m \in \mathbf{m}}
\]
Are we good now?

• Defects
  • Write through pointers
  • Indirect call instructions
  • Read through pointers

• Self-Integrity Compiler!
  • Bitmap
  • Dynamic Check
Extension & Discussion

• How does the system guarantee confidentiality and integrity?

• How does system guarantees verifiability?

• Can we launch Meltdown or Spectre attack SGX?

• We need to trust intel to trust the quote scheme, so is this scheme just a transfer of trust? (Like many other schemes)