Review “VC3”

Name: Wenhui Zhang

Paper Name：

Schuster, F., Costa, M., Fournet, C., Gkantsidis, C., Peinado, M., Mainar-Ruiz, G., & Russinovich, M. (2015, May). VC3: Trustworthy data analytics in the cloud using SGX. In *Security and Privacy (SP), 2015 IEEE Symposium on* (pp. 38-54). IEEE.

Contribution:

VC3 provides shielded execution of distributed MapReduce computations in the Cloud, while cloud providers are NOT trusted. It guarantees: (1) Confidentiality of computation and results; (2) Isolation of map and reduce computations; (3) Integrity of data, computation, and code.

Motivation:

Cloud users require security guarantees of (1) Confidentiality and integrity for both code and data; (2) Verifiability of execution of the code over the data; i. E., the guarantee that their distributed computation globally ran to completion and was not tampered with.

Related works:

Fully homomorphic encryption[Gentry, STOC 2009] not work for shared computation; CryptDB and Cipherbase protection for partially for practical usage, not complete mediation.

Methodology:

VC3 MapReduce computation performs Map() or Reduce() function inside SGX enclaves. Map() or Reduce() Code are compiled into a self-contained binary, and are running in SGX enclaves. Code are written in C++, and CFI is enforced (two realization: region write integrity and region write and read integrity). The communication channel between the client and an SGX enclave is attestented through signatures, and private data retrieval is through this channel.

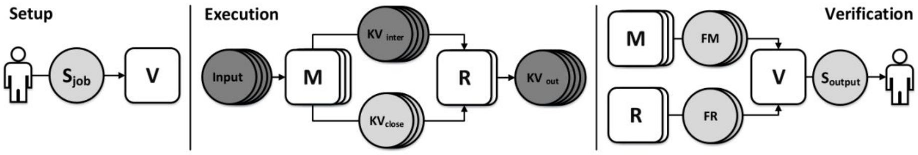

Figure 3: Schematic overview of our job execution protocol. The verifier (**V**), mappers (**M**), and reducers (**R**) are depicted as squares. A light-gray circle displays a message/key-value pair that is sent once by an entity; a dark-gray circle one that is sent multiple times. The user is depicted at both far ends. *Image and caption taken from Schuster et. al [3].*

Results:

Intermediate results sent from mappers to reducers are encrypted. They are in key-value pairs format. Keys (bins) are obfuscated using a one-way hash and values are encrypted using an intermediate key. Because the key-value pairs could be dropped or replayed, a verifier checks whether each pair is processed once and once only.

Take away:

A particular vulnerability remains despite the utilization of SGX enclaves; unsafe memory access while running in enclaved code. The authors address this issue by performing code analysis in their compiler toolchain to prevent unsafe memory access. The paper does not address DoS attacks, side-channels, or traffic sniffing.