University of Hong Kong

Final Year Project

Building Secure Big-data Application

Interim Report

CHEN Xing Quan Hansen (3035440912)
Department of Computer Science

Supervised by

Dr. Heming Cui
Department of Computer Science

February 2, 2019

Contents

1 Introduction

Nowadays cloud computing paradigm enables various big data applications (e.g., Spark [1]) to be deployed in public clouds. Thus, sensitive data suffers from severe security threats in public clouds. Some of these applications are written in Java including Apache Storm. Although Java has many security features including type-safety, the JVM runtime alone is insufficient to defend against privileged attacks, because adversaries may control the entire cloud software stack, including OS kernels [2], [3].

Trusted Execution Environment (TEE) is a promising technique for protecting sensitive data for data-handling applications in public clouds, and the Intel Software Guard eXtensions (SGX) is a major TEE product. SGX provides an enclave execution abstraction with limited memory (around 100MB). SGX runs trusted code in an enclave and uses the CPU hardware to prevent attackers.

This final year project includes applying Intel SGX to a popular big data streaming software, Apache Spark.

The following of this report is structured in this way: Section 2 provides some background of the sensitive data protection technique and big data software used in this final year project including Intel SGX and Storm; Section 3 presents the objective of this project; Section 4 discusses the methodologies applied in this project; Section 5 reports the challenge in this project as well as the project progress and Section 6 shows the detailed project schedule.

2 Background

This section provides an overview of significant terms mentioned in Section 1 and the background knowledge needed for building secure big data application.

2.1 Intel SGX

An SGX enclave isolates the execution environment of the application code and data running inside and protects them from outside privileged access, including OS, hypervisor, and BIOS. Memory pages belonging to an enclave reside in the Enclave Page Cache (EPC). EPC has a total size of 128MB per CPU, and only around 100MB can be used by application code. If the code running in the enclave uses more than 100MB, a slow SGX paging mechanism will incur a 1,000X slowdown compared to regular OS paging [4]. An SGX enclave can execute only user-space instructions, so the enclave code has to do OCalls to leave enclaves for system calls. When an interrupt or hardware exception (e.g., General Fault) is raised in an enclave, the processor performs an Asynchronous Enclave Exit (AEX) to handle the exception or interrupt. Though AEX and OCall do not directly leak secrets in enclaves, they can result in severe side-channels or even attack surface for revealing the control flow of trusted code and plaintext data.

2.2 Apache Storm

Apache Storm is a free and open source distributed real-time computation system. It enables reliably process of unbounded streams of data, doing for real-time processing similar to Hadoop batch processing. Also, Apache Storm supports may programming languages including Python, Java, etc. Thus, Apache Storm has many use cases including real-time analytics, online machine learning, continuous computation, distributed RPC, ETL, etc. In addition, Apache Storm is clocked by a benchmark at over a million tuples processed per second per node [5]. The main computation part of Apache Storm is considered as a topology. An Apache Storm topology consumes streams of data and processes those streams in arbitrarily complex ways, repartitioning the streams between each stage of the computation however needed.

3 Objective

3.1 Scope

Considering the complexity of conducting research and the engineering work of research project, the scope is temporarily decided to the following and is subject to changes in the actual engineering work and research progress:

Application of Intel SGX Function Annotation in Apache Storm. Apache Storm is a distributed real-time computation system used by many big companies including Yahoo, Twitter, etc [5]. It provides easy functionalities to reliably process unbounded streams of data, i.e. real-time processing. However, a privileged attacker may try to see sensitive data or to tamper with application logic by OS and debug attacks. Therefore, Intel SGX is planned to be implemented to annotate sensitive functions to let the computation of Storm Bolt happen inside enclave protection using ECall and the I/O between Storm Bolt happen as OCall from inside enclave to outside enclave.

Implementation of Encryption and Decryption inside Intel SGX Enclave. After annotating the sensitive functions, the input and output of Storm Bolt may also be acquired by attackers to interpret the computation inside enclave. Thus, decryption will be performed when Storm Bolt receive input and encryption will be performed when Storm Spout send input to Storm Bolt and Storm Bolt send intermediate result to other Storm Bolt. Both encryption and decryption are performed inside enclave with algorithms developed by PhD student of my supervisor and details will be given in final report.

3.2 Deliverable

This project will deliver:

i) effective application of Intel SGX Function Annotation in Apache Storm with acceptable performance;
ii) successful implementation of encryption and decryption inside Intel SGX enclave
iii) evaluation of enhanced secure big-data streaming application, i.e. storm with Intel SGX and encryption/decryption

The project progress can be checked on the website: i.cs.hku.hk/fyp/2019/fyp19032

4 Methodologies

In order to make sure this final year project is delivered successfully, we divided the project into three main phases and the methodology for each phase is:

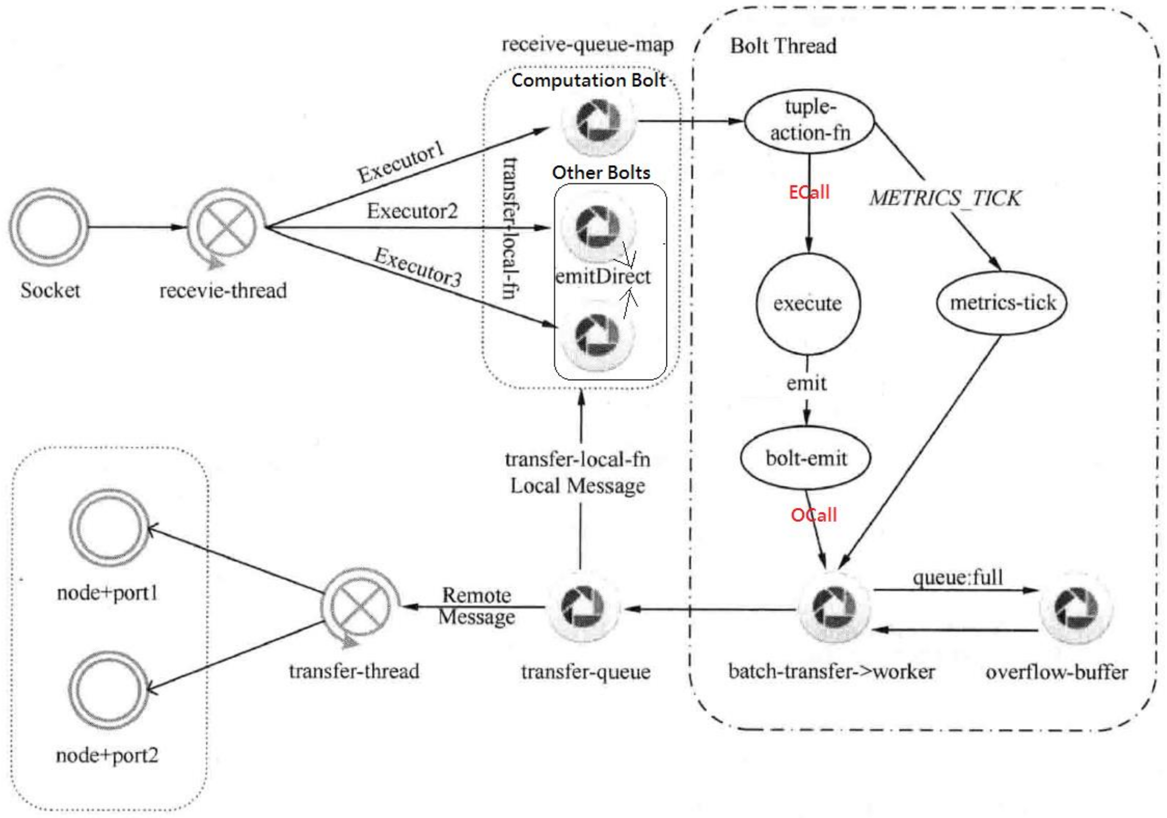
Detailed Partition Design. For the implementation of Intel SGX Function Annotation in Apache Storm, detailed functions partition is needed because the source code is developed in a compact way such that many component share the same function including the function need to be annotated. Thus, duplication of some original functions are performed with slightly difference for entering enclave(ECall) and I/O happened from inside enclave to outside enclave(OCall).

Engineering Work. Proper engineering works including but not limited to VM setup, server configuration, Java and Intel SGX programming are conducted in the computers provided by supervisor and PhD student. Engineering works are expected to be conducted in the source code of Apache Storm in low level manner according to the detailed partition design. Intel SGX technique is expected to be integrated into the computation, encryption/decryption as well as I/O related functions of Apache Storm.

Evaluation. After the successful engineering work, evaluation of the secure big data application compared to the original big data application or other existing secure big data systems running the same computations will be conducted. The metric for performance tentatively includes but is not limited to: tuples processed per second per node.

5 Challenges and Mitigations

One major challenge encountered in this project is that the computation bolt containing client defined computation and other bolt including SystemBolt, etc. used for coordinating share the same execution function “execute” and I/O function “emit/emitDirect”. The current solution is to direct the computation bolt to annotated execute function entering enclave and other bolts to normal execute function to reduce trust code base and duplicate the main function “bolt-execute” used for emit and emitDirect and only let computation bolt use OCall annotated bolt-emit function and other bolt (e.g. Acker bolt uses emitDirect) calling normal bolt-emit function. The detailed visualization is presented below with red words indicating annotation.



The other challenge is that performing I/O between bolts needs to copy the output and the transfer needed objects including ExecutorTransfer etc. out of the enclave. This will affect the performance of computation as deep copy consumes more time than only performing I/O. The current solution is to choose as least objects as possible to copy out of enclave memory. For example, we instead of copying the entire bolt executor out of enclave, we may only copy needed variables inside bolt executor out of enclave for performing I/O.

6 Schedule

Time Periods	Tasks
September	• Meeting with Supervisor • Project Plan Writing • Project Website Design • Literature Review
October	• Literature Review • System Testing for Original Apache Storm • Development of the first version of Intel SGX Annotation implemented Storm
November - December	• Complete the Detailed Function Partition Design • Start to Implement the Second Version Annotation according to Partition Design • Interim Report Writing
January - March	• Annotation implementation with Correctness and Acceptable Performance • Encryption and Decryption Implementation with Acceptable Effect to the Performance
March - April	• System Evaluation • Final Report Writing • Final Presentation • Poster Design

7 Conclusion

This is an interim report for my final year project. As more and more commercialized big data applications are deployed to public cloud, security concerns have been in the public eye. As more and more related researches are conducted, proper engineering works have to be conducted to make existing big data streaming application more secure. Therefore, we try to build secure big data streaming application upon research product, i.e. apply Intel SGX to Apache Storm. We will also implement encryption/decryption, fine-tune the performance of Intel SGX implemented Apache Storm and evaluate the system.

The project website is: https://i.cs.hku.hk/fyp/2019/fyp19032/

References

[1] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, “Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing,” in NSDI, pp. 2–2, USENIX Association, 2012.
[2] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, “Vc3: Trustworthy data analytics in the cloud using sgx,” in Security and Privacy (SP), 2015 IEEE Symposium on, pp. 38–54, IEEE, 2015.
[3] W. Zheng, A. Dave, J. G. Beekman, R. A. Popa, J. E. Gonzalez, and I. Stoica, “Opaque: An oblivious and encrypted distributed analytics platform.,” in NSDI, pp. 283–298, 2017.
[4] S. Brenner, C. Wulf, D. Goltzsche, N. Weichbrodt, M. Lorenz, C. Fetzer, P. R. Pietzuch, and R. Kapitza, “Securekeeper: Confidential zookeeper using intel sgx.,” in Middleware, p. 14, 2016.
[5] Apache Storm [n.d.]. Apache Storm. https://storm.apache.org/