Safeguarding Data Consistency at the Edge

Cláudio Correia
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
claudio.correia@tecnico.ulisboa.pt

***Abstract*—We propose to design and implement a secure edge storage system. Edge computing is a paradigm that extends cloud computing with storage and processing capacity close to the edge of the network, supporting new applications that require low latency. It assumes the availability of fog nodes that are located close to the edge. However, fog nodes are likely to be vulnerable to tampering. A malicious fog node can manipulate, create or delete data from edge applications, leading these applications into a fail state, impacting the quality of service. Therefore, it is important to secure the functions fog nodes provide.**

To achieve our goal we plan to leverage the use of secure hardware (e.g., Intel SGX) as a means to harden the implementation. However, as we discuss here, SGX alone is not enough to achieve the qualities we consider necessary to support edge applications, such as low latency, scalability, and multiple models of data consistency. In this work, we present the main challenges in the design of a secure edge storage system and point to the research directions that we plan to follow to address these challenges.

***Index Terms*—Security, IoT, Fog, Edge, Intel SGX**

I. Introduction

Many applications deployed in the cloud provide a range of services to clients that reside in the edge of the network, also known as the Internet of Things (IoT). Edge devices often run time-constrained applications, such as augmented reality or online games, that require low latency when accessing the cloud. In fact, a response time below 5ms–30ms is typically required for many of these applications to be usable [1].

Edge computing is a model of computation that aims at leveraging the capacity of fog nodes to run part of the computation typically done in the cloud, saving network bandwidth and providing results with low latency. Fog nodes are most likely managed by several different local providers and installed in physical locations that are more exposed to tampering. Therefore, fog nodes are substantially vulnerable to be compromised [2], and developers of applications and middleware for edge computing need to take security as a primary concern. A promising approach to ease application development in this setting is to offer secure middleware components, such as secure storage services, that can shield the application from the fog nodes vulnerabilities.

In our work we focus on edge storage services, that extend those offered by the cloud in such a way that relevant data is replicated closer to the edge. More precisely, we aim at designing and implementing a secure storage system that uses fog nodes as data replicas. We plan to leverage the use of secure hardware (e.g., Intel SGX) as a means to harden the implementation of our storage service. In this work, we identify the main challenges that emerge when one attempts to build an edge storage service based on Intel SGX.

II. Research Objectives

Our main objective is to design and implement a secure storage system for the edge. The system must ensure the following properties, that we have identified as relevant for most edge applications: offer *low latency* for edge clients; be *scalable* (in terms of number of clients served and amount of data it can store) and; offer multiple *data consistency guarantees* (e.g. causal and total order).

III. Approach

To achieve our goals, our storage system has to take advantage of fog nodes as storage replicas, thus achieving lower latencies. However, as mentioned earlier, fog nodes are vulnerable to attack and can lead our system to a state of failure, with serious consequences on the quality of service on edge applications. A promising approach is the use of Intel Software Guard Extensions (SGX) in fog nodes. Intel SGX implements a secure execution mode called an enclave. The enclave can guarantee the confidentiality and integrity of data and code that runs on the processor, even if the processor is under attack, preventing edge applications from violating safety constraints. The use of SGX is a topic that has deserved substantial attention from researchers in the recent past [3], [4], and several limitations and attacks have already been identified. These limitations raise a number of challenges in the design of storage services, that we strive to overcome.

IV. Challenges of a Storage System at the Edge

In our work, we assume that all fog nodes/storage replicas have an Intel SGX processor. With the use of SGX, we have a base of trust in all fog nodes even if they are attacked [2], the enclave is a secure entity even if the operating system is compromised. With the use of SGX, fog nodes are less vulnerable, however challenges still arise. Below we list the three main properties that an edge storage system must hold and we present the challenges that each brings.

Scalability: A limitation of enclaves is the size of their memory space (≈120MB). Previous storage systems [3]–[5] have solve this using cryptographic techniques to store data outside the enclave in a secure way. However, these systems make compromises between scalability and latency: the storage capacity is limited by the metadata size they can store inside the enclave or they use techniques to validate the integrity of data outside the enclave that exhibits a search cost that grows linearly with the storage capacity. Thus, previous work suffers from scalability limitations that need to be circumvented to make the storage service practical.

978-1-7281-7260-6/20/$31.00 ©2020 IEEE
DOI 10.1109/DSN-S50200.2020.00035

Low latency: A storage system that takes advantage of fog nodes for replication has the potential to respond to edge clients with low latency. However, on one hand, the use of intel SGX can introduce new sources of latency and, on the other hand, it does not prevent rational or malicious behavior that violates latency constraints. We have identified the following latency-related challenges.

First, most systems based on SGX require that every operation goes through the enclave, incurring in a non-negligible latency overheard. Techniques need to be devised to avoid calling the enclave at every invocation.

Second, previous systems also require clients to perform the attestation of the enclave, every time they interact with a new fog node. This attestation procedure, whose purpose is to ensure that the client is interacting with an enclave that has not been compromised, incurs in a high latency penalty, as it implies contacting Intel services. Intel DCAP is a recently proposed approach to speedup attestation, but still requires clients to store multiple files or communicate with a trusted third identity. More efficient attestation services are needed.

Third, malicious or rational fog nodes may opt not to offer low latency, in order to obtain benefit. For instance, an opportunistic service provider can store all data in the cloud (where memory is cheaper) instead of storing it in the fog node; this provider can still satisfy client request (by fetching data from the cloud) but will fail to provide the low latency expected by clients. Mechanisms to detect this sort of rational behavior need to be devised.

Data consistency: Similarly to what happened in the cloud, storage systems will evolve to have multiple replicas and relax data consistency in exchange for performance. Even weak consistency models, such as causal consistency, require coordination to execute a client request in a safe manner. This coordination raises challenges at the level of individual nodes, and at the level of the cooperation among distributed nodes. We identify the following challenges:

First, in the current SGX architecture, it is possible to undetectably launch multiple instances of the same enclave within a single machine. This is a thread to consistency, in particular when different calls to the trusted service need to be synchronized and, therefore, processed by a single enclave instance. To the best of our knowledge, only Rote [6] has attempted to solve this challenge using multiple communications, a solution to be avoided due to the cost of latency.

Second, all previous storage systems using SGX are designed to work in as centralize manner, leaving open the challenge of synchronization of multiples replicas, supported by enclaves, to offer data consistency.

V. Research Directions

We now present directions to address the above challenges.

Scalability: None of the presented systems [3]–[5] allows data to be stored locally in a fog node in a scalable manner with low latency. It is important to find a combination of efficient techniques for data stored as a cache inside the enclave and for data outside the enclave (e.g., Merkle Trees).

Low latency: Using cryptographic techniques is possible to use the enclave as a root of trust for just a few important operations, leaving other operations to be performed without the enclave intervention. Techniques such as logs, blockchain, graphs or homomorphic encryption can be a solution so that a client can perform reading operations without the intervention of the enclave (while ensuring integrity and authenticity). Edge clients can obtain trust in the enclave without performing any type of attestation, solutions like Excalibur [7] are preferable to be used at the edge. Finally, a storage system requires techniques that guarantee that the data is indeed replicated at the edge and not just in the cloud (e.g., proof of storage).

Data consistency: A local solution is desirable to prevent the multiple enclave instances attack. Potential solutions include the use of unique tokens or a special enclave responsible for initializing other enclaves. It is important to design protocols for communication between multiple enclaves. Synchronization techniques allowing enclaves to guarantee data consistency (e.g. causal or total order) for writing, reading, and even remote updates.

— As a final step, we will combine all the previous techniques in one distributed system and evaluate its performance and limitations.

VI. Preliminary Results

As a first step in the execution of our research plan, we have designed and implemented Omega [8]. Omega takes advantage of Intel SGX and uses a Merkle tree to store data outside the enclave in a secure and scalable approach. However, Omega does not take advantage of the enclave memory as a cache to store data. Omega is designed considering a single replica, leaving open many of the challenges mentioned above. We learned from Omega that the overhead of using the enclave is minor, hence the use of enclaves is a suitable solution for the edge. We also observed that digital signatures are the major reason of overhead, meaning, we need structures that avoid the constant use of the enclave to achieve lower latencies.

Acknowledgments: This work was partially supported by the Fundação para a Ciência e Tecnologia (FCT) via project UIDB/50021/2020 (INESC-ID) and by the European Commission under grant agreement number 830892 (SPARTA).

References

[1] G. Ricart, "A city edge cloud with its economic and technical considerations," in *Workshop on SmartEdge*, Kona, HI, USA, Jun. 2017.
[2] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, no. 6, 2017.
[3] T. Kim, J. Park, J. Woo, S. Jeon, and J. Huh, "Shieldstore: Shielded in-memory key-value storage with SGX," in *EUROSYS*, Dresden, Germany, Mar. 2019.
[4] M. Bailleu, J. Thalheim, P. Bhatotia, C. Fetzer, M. Honda, and K. Vaswani, "Speicher: Securing LSM-based key-value stores using shielded execution," in *FAST*, Boston, MA, USA, Feb. 2019.
[5] L. Chen, J. Li, R. Ma, H. Guan, and H.-A. Jacobsen, "EnclaveCache: A secure and scalable key-value cache in multi-tenant clouds using Intel SGX," in *Middleware*, Davis, CA, USA, Dec. 2019.
[6] S. Matetic, M. Ahmed, K. Kostiainen, A. Dhar, D. Sommer, A. Gervais, A. Juels, and S. Capkun, "ROTE: Rollback protection for trusted execution," in *USENIX Security*, Vancouver, Canada, Aug. 2017.
[7] N. Santos, R. Rodrigues, K. P. Gummadi, and S. Saroiu, "Policy-sealed data: A new abstraction for building trusted cloud services," in *USENIX Security*, Bellevue, WA, USA, Aug. 2012.
[8] C. Correia, M. Correia, and L. Rodrigues, "Omega: a secure event ordering service for the edge," in *DSN*, València, Spain, Jun. 2020.