Confidential Cloud Services

Neele Peter
neele.peter@fau.de
Friedrich-Alexander-Universität Erlangen-Nürnberg

ABSTRACT

TODO

1 INTRODUCTION

A Confidential Cloud Service is an additional service that can be applied on top of an existing cloud provider [8]. These confidential services are needed, if properties have requirements of safety, which cannot be guaranteed by a normal cloud provider.
The CIA triad, shown in Figure 1, describes three important requirements of information security [1, 11]. It contains data Confidentiality, Integrity Protection and High Availability.
Data confidentiality is keeping the data private, which is very important especially for cloud providers. Challenges are encrypting the data and protecting the keys.
Integrity protection is introduced as a requirement for data confidentiality. It is the insurance of complete and correct code that is not changed by a bad party. But these two characteristics are hard to implement although they are mandatory for cloud computing. This is, because in cloud computing the trusted computing base (TCB) gets bigger and also includes the cloud providers, so applications in the field of finances, medicine or governmental issues cannot afford to trust this whole TCB.
High availability is required by the fact that people rely on the systems that are on the cloud. So they should work, even if failures occur.

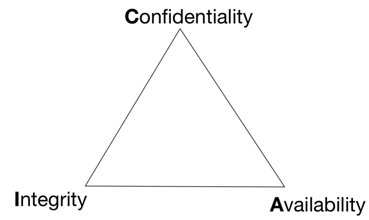

Figure 1: Graphic of the CIA triad

2 BACKGROUND

Before I start with an example of a confidential cloud service some definitions must be clear to understand the following parts.

2.1 Trusted Execution Environments

A Trusted Execution Environment (TEE) is a hardware based component where critical code can be runned inside and is stored in a part called enclave. The code inside this Enclave is hard to modify by malicious parties and can thus be trusted more and is therefore often be used for confidential computing. But of course it is not impossible to change code inside a TEE, so this eventualities should also be kept in mind. Through the concept of remote attestation it is possible to make sure that the correct code runs inside the TEE [3].
Trusted Execution Environments can be realized either on a hardware processor like Intel SGX [6] or in a virtual machine like AMD SEV-SNP [10].

2.2 Ledger

A ledger is a digital register that is made to document transactions or other structural data. It is often used for blockchains or confidential computing.

2.3 Rollback Attacks

There exist many different attacks against cloud services like forking attacks [4], side channel attacks [5] or rollback attacks. These are all important attacks that should be detected and protected by a confidential cloud service, but in this paper I will focus on rollback attacks [7].
In this attack malicious parties safe an older version of the system, restart it, and apply this older version as a new state. So they "roll back" the state of the system. This can be useful for guessing encryption keys. For these keys there often exist some limitations of guesses, so the key cannot get brute forced. The problem is with rolling back the state of the system a malicious member can go back to the state before the first try and guess again. So brute forcing the encryption key is still possible by using these attacks.

3 CCF

One example of a service, that can be added on a cloud to make it confidential is the Confidential Consortium Framework (CCF) [9]. It wants to guarantee the CIA requirements on multiparty applications.

3.1 Concept

The basic idea of CCF is that an untrusted host exists and one or more untrusted users can access different replicated nodes that are responsible for the remaining communication. One of these nodes is the primary. Users can connect to any node and the specific request is either being forwarded to the primary or handled by the node itself. Operations that do not have the obligation to be handled in the primary are read-only requests. If this node fails the user can be connected to another node, if the primary fails there is a primary election that elects a new primary with certain voting criteria. The primary also periodically sends signature transactions. These are transactions that confirm that there were no malicious changes to the code and thus guarantee the integrity of CCF. Transactions are provided via a merkle tree, where each transaction is a hash value. Every two nodes in the merkle tree are combined with a hash function to a new node. On top of the merkle

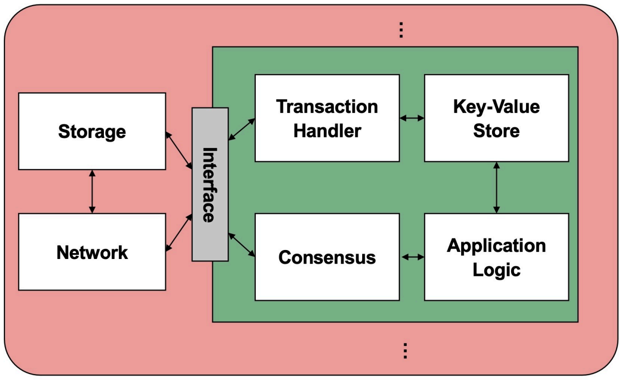

Figure 2: Each node consists of a TEE, the untrusted host, and an interface between the host and the TEE. The TEE consists of a Key-value store, the Consensus, the Transaction Handler and the specific Application logic.The storage and the network are placed outside the TEE.

tree is the merkle root, which is the value that is used for the signature transaction.Providers only need to implement the application logic and make sure to have all necessary endpoints, so that CCF can handle these. Users then only have to access the specific endpoints to use the application. In Figure2 the structure of one node is described. The application logic gets the data from the Key-value-store which gets the data from the Consensus. The Transaction Handler stores every signature transaction in an append-only ledger that is redundant in every other node and the persistent storage. Performing the application logic inside a TEE is fundamental for the confidentiality and the integrity, replicating the transactions is necessary for the high availability. The ledger is stored outside the TEE what leads to certain problems that are discussed in 3.6.

3.2 Reconfiguration

Reconfiguration is the procedure when a node fails and is replaced by a new one. This is an important feature to guarantee the high availability. CCF therefore allows to add new or delete old nodes. Reconfiguration is implemented as a transaction. For Reconfiguration a node must request an election and win it. The election is done by a majority quorum.

3.3 Desaster Protocol

3.4 Confidentiality

3.5 Rollback Attacks

3.6 Challenges

As mentioned before CCF does not have a rollback detection, because it allows rollback attacks to happen, because it is not affecting the system. The problems are that they (1) make the assumption that the code inside the TEE cannot be changed and (2) that thy put their persistent storage outside the TEEs and is therefore not protected for a rollback attack.

Solutions for these challenges already exist. In the following section I introduce another existing confidential cloud service that especially focuses on detecting and protecting rollback attack.

4 ROLLBACK PROTECTION

Nimble[2] also tries to ensure the requirement of the CIA triad, but has the focus of preventing rollback attacks. In the following section the functions of Nimble are introduced and important components are compared with the ones of CCF.

4.1 Concept

Nimble has three important goals, (1) ensure linearizability, (2) having a small Trusted Computing Base (TCB) and (3) to guarantee liveness of the cloud service. Therefore the authors differ between two main properties:

- **Safety** means that it must be ensured that every data that can be accessed must be the current data and must not be an older version. This property is enabled via TEEs. (In the CIA triad this would be Confidentiality and Integrity)
- **Liveness** is the High Availability property of the CIA triad. This has not to be ensured via a TEE, because liveness can be easily taken away even in an TEE, and can be handled outside what simultaneously makes the TCB smaller.

To guarantee liveness Nimble stores the data redundantly via state machine replication. The state of each replicated node is apended to a ledger which is stored as a hash chain in an existing storage provider. To guarantee safety the state is encrypted and also stored in an append-only leger.

4.2 Rollback Attacks

To protect rollback attacks they firstly must be detected. Therefore the paper presents three categories that all detect a rollback attack:

- **Addressing stale responses:** Stale responses happen when old data is given by the host although there is already newer data. Therefore Nimble uses an linearizable append-only ledger. Linearizability is a criteria that provides strong consistency.
- **Addressing synthesized requests:** Synthesized requests means that the provider sends requests that are not sent by the application and stores them. This is handled via signing. The application signs the state and then appends it on the ledger. When the application reads a state from the ledger it then can recognize whether it is a real state with a signature or a synthesized request from a malicious provider.
- **Addressing replay:** When a provider applies older requests to the storage this is called replay. Therefore the position of the stored state in the data is also stored with a signature. So the application can check, if this signed position matches with the position it is stored in the ledger.

4.3 Confidentiality

4.4 Reconfiguration

4.5 Desaster Protocol

4.6 TCB

4.7 Challenges

5 RELATED WORK

5.1 Services without TEEs

simple key-value store

5.2 Services with TEEs

6 CONCLUSION

REFERENCES

[1] Michael Aminzade. Confidentiality, integrity and availability–finding a balanced it framework. *Network Security*, 2018(5):9–11, 2018.

[2] Sebastian Angel, Aditya Basu, Weidong Cui, Trent Jaeger, Stella Lau, Srinath Setty, and Sudheesh Singanamalla. Nimble: Rollback protection for confidential cloud services (extended version). Cryptology ePrint Archive, Paper 2023/761, 2023. https://eprint.iacr.org/2023/761.

[3] Alexander Sprogø Banks, Marek Kisiel, and Philip Korsholm. Remote attestation: A literature review. *CoRR*, abs/2105.02466, 2021.

[4] Samira Briongos, Ghassan Karame, Claudio Soriente, and Annika Wilde. No forking way: Detecting cloning attacks on intel sgx applications. In *Proceedings of the 39th Annual Computer Security Applications Conference*, ACSAC '23, New York, NY, USA, 2023. Association for Computing Machinery.

[5] Sebanjila Kevin Bukasa, Ronan Lashermes, Hélène Le Bouder, Jean-Louis Lanet, and Axel Legay. How trustzone could be bypassed: Side-channel attacks on a modern system-on-chip. In *Information Security Theory and Practice: 11th IFIP WG 11.2 International Conference, WISTP 2017, Heraklion, Crete, Greece, September 28–29, 2017, Proceedings 11*, pages 93–109. Springer, 2018.

[6] Victor Costan and Srinivas Devadas. Intel SGX explained. Cryptology ePrint Archive, Paper 2016/086, 2016. https://eprint.iacr.org/2016/086.

[7] Levente Csikor, Hoon Wei Lim, Jun Wen Wong, Soundarya Ramesh, Rohini Poolat Parameswarath, and Mun Choon Chan. Rollback: A new time-agnostic replay attack against the automotive remote keyless entry systems. *ACM Trans. Cyber-Phys. Syst.*, 8(1), jan 2024.

[8] Sascha Fahl, Marian Harbach, Thomas Muders, and Matthew Smith. Confidentiality as a service–usable security for the cloud. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 153–162. IEEE, 2012.

[9] Heidi Howard, Fritz Alder, Edward Ashton, Amaury Chamayou, Sylvan Clebsch, Manuel Costa, Antoine Delignat-Lavaud, Cedric Fournet, Andrew Jeffery, Matthew Kerner, Fotios Kounelis, Markus A. Kuppe, Julien Maffre, Mark Russinovich, and Christoph M. Wintersteiger. Confidential consortium framework: Secure multiparty applications with confidentiality, integrity, and high availability, 2023.

[10] AMD Sev-Snp. Strengthening vm isolation with integrity protection and more. *White Paper, January*, 53:1450–1465, 2020.

[11] Michael E Whitman, Herbert J Mattord, et al. *Principles of information security*. Thomson Course Technology Boston, MA, 2009.