Cloud-aware BFT Proactive Recovery Using Confidential Computing

Ines Messadi and Rüdiger Kapitza
Friedrich-Alexander-Universität Erlangen-Nürnberg

Abstract

Byzantine Fault Tolerance (BFT) systems provide strong integrity guarantees by tolerating arbitrary faults, given that a threshold of participants remain non-faulty. Sustaining such a threshold in a cloud environment poses a challenge as cloud providers have control over the nodes and infrastructure. Even though some degree of resource availability has to be guaranteed by the provider, organizations are usually forced to place their trust in the cloud to respect the integrity and confidentiality of their data. To address the latter, Trusted Execution Environment (TEE) can be utilized to ensure data confidentiality and integrity while diminishing the level of trust required. However, software running inside TEEs is not flawless, especially in long-lived BFT systems that require updates to support software evolution.

In this talk, we explore the concept of proactive recovery and rejuvenation of BFT systems in the cloud as a strategy to eliminate hidden faults in TEEs and enable upgrades. We present a cloud-ready system featuring a proactive recovery-aware protocol that builds on fundamental TEE mechanisms.

1 Problem Statement

Distributed ledgers and permissioned blockchains are popular because they securely handle diverse transactions, such as medical data and supply chain [1, 10]. BFT state machine replication forms the backbone of recent permissioned blockchains due to its ability to deliver high throughput and ensure applications' integrity, provided the majority of participants are non-faulty [4, 5]. Originally, BFT systems were primarily on-premise, operating on an infrastructure that is locally managed and owned by a single party. Recognizing the complexity of setting up a BFT blockchain, multiple providers now offer a user-friendly cloud-based variant to expand their product offering [2, 9]. While this approach enhances accessibility, it puts the provider in a central position for confidentiality, integrity through the access to the replicas and infrastructure. This situation can lead to concerns, as the involvement of the cloud operators opens doors for potential insider attacks, such as rogue administrators.

Confidential computing minimizes the need to trust the provider staff by utilizing a TEE for computations beyond the control of hosting parties. Deployments of TEE have proven beneficial for commodity workloads, enhancing applications' security in the untrusted cloud. Indeed, first replicated systems leverage TEE for cloud deployment [3, 8, 11]. However, these systems feature a hybrid system model where the replication logic is placed inside the TEE, considered flawless and only fails by crashing. While seemingly robust, this design contributes to the system's Trusted Computing Base, potentially increasing the risk of vulnerabilities over time. Prior research in TEE shows that certain application bugs or memory corruptions could potentially cause an exploitable vulnerability [7, 12]. These systems, typically expected to function over extended periods, can become vulnerable to targeted attacks and safety breaches due to a single flaw. Over time, these faults can accumulate, leading to inconsistencies. In conclusion, hybrid models are not well-suited for long-lived systems as faults should be flashed out to maintain integrity.

Traditional methods, such as simply restarting the TEE, are insufficient as they fail to eliminate the attacker who might exploit the same code vulnerability again. A prevalent approach to mitigate such faults involves proactively recovering the TEE node at regular intervals using rejuvenation techniques such as security patches and code randomization. This approach aims to maintain the integrity of the service throughout its entire operational lifespan. Importantly, the integrity of a BFT service running in the cloud should not depend on a cloud provider, including system upgrades. The foundational system, BFT-PR [6], introduces the concept of recovery. However, it relies on hardware requirements such as watchdogs, which are not typically provided by commodity cloud infrastructures. This results in a lack of compatibility with cloud environments. In addition, while BFT-PR does not account for upgrades, in a cloud setting, upgrades must be provable and integrity preserved following each update. Currently, no system is ready for the cloud and supports a proactive recovery

approach for permissioned blockchains.

2 TEE-based Recovery Approach

In this talk, we introduce a system that addresses these problems through *a TEE recovery-aware protocol* to enhance the practicality of TEE-based BFT systems in the cloud. Our approach relies on a recovery process to evict potential attackers that may accumulate over the system's lifetime; this involves creating a distinctly different new replica instance. We enable recovery through a distributed attestation, which includes a proof of a clean start for new nodes, and we account for an easily upgradable system to better accommodate real-world settings.

References

[1] Building a Transparent Supply Chain. https://hbr.org/2020/05/building-a-transparent-supply-chain. Accessed: 2024-02-04.

[2] AMAZON. Amazon Managed Blockchain. https://aws.amazon.com/managed-blockchain/, 2024.

[3] BAILLEU, M., GIANTSIDI, D., GAVRIELATOS, V., NAGARAJAN, V., BHATOTIA, P., ET AL. Avocado: A secure {In-Memory} distributed storage system. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)* (2021), pp. 65–79.

[4] BARGER, A., MANEVICH, Y., MEIR, H., AND TOCK, Y. A byzantine fault-tolerant consensus library for hyperledger fabric. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (2021), IEEE, pp. 1–9.

[5] BERGER, C., SCHWARZ-RÜSCH, S., VOGEL, A., BLEEKE, K., JEHL, L., REISER, H. P., AND KAPITZA, R. Sok: Scalability techniques for bft consensus. *arXiv preprint arXiv:2303.11045* (2023).

[6] CASTRO, M., AND LISKOV, B. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS) 20*, 4 (2002), 398–461.

[7] CLOOSTERS, T., RODLER, M., AND DAVI, L. Teerex: Discovery and exploitation of memory corruption vulnerabilities in {SGX} enclaves. In *29th {USENIX} Security Symposium ({USENIX} Security 20)* (2020), pp. 841–858.

[8] HOWARD, H., ALDER, F., ASHTON, E., CHAMAYOU, A., CLEBSCH, S., COSTA, M., DELIGNAT-LAVAUD, A., FOURNET, C., JEFFERY, A., KERNER, M., ET AL. Confidential consortium framework: Secure multiparty applications with confidentiality, integrity, and high availability. *arXiv preprint arXiv:2310.11559* (2023).

[9] IBM. IBM Blockchain Services. https://www.ibm.com/blockchain, 2024.

[10] KUO, T.-T., KIM, H.-E., AND OHNO-MACHADO, L. Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association 24*, 6 (2017), 1211–1220.

[11] WANG, W., DENG, S., NIU, J., REITER, M. K., AND ZHANG, Y. Engraft: Enclave-guarded raft on byzantine faulty nodes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (2022), pp. 2841–2855.

[12] WEICHBRODT, N., KURMUS, A., PIETZUCH, P., AND KAPITZA, R. AsyncShock: Exploiting Synchronisation Bugs in Intel SGX Enclaves. ESORICS.