TEE-KV: Secure Immutable Key-Value Store for Trusted Execution Environments

Atsushi Koshiba[1], Ying Yan[2], Zhongxin Guo[3], Mitaro Namiki[1], and Lidong Zhou[3]

[1] *Tokyo University of Agriculture and Technology* [2] *Ant Financial* [3] *Microsoft Research Asia*

CCS CONCEPTS

• **Security and privacy → Database and storage security**; *Trusted computing*;

KEYWORDS

key-value store, trusted execution environments, blockchain

ABSTRACT

Trusted Execution Environments (TEEs) ensure strong data confidentiality for applications running in the TEEs even on untrusted servers. In particular, TEEs are expected to bring significant benefits to blockchain workloads for enterprise because it ensures confidentiality and correctness of transaction records without any heavy-weight data verification process such as proof-of-work. For example, Coco [3] improves both confidentiality and transaction throughput of existing blockchain protocols by utilizing TEE features.

Although executing blockchain workloads in TEEs is a promising technique, applying databases such as key-value store (KVS) to TEEs is challenging. Databases are essential for blockchain systems to manage a large amount of transaction records. However, TEEs are not suitable for running databases due to their limitations. First, since databases are memory-intensive and large-scale workloads, putting them into TEEs increases Trusted Computing Base (TCB) size, which makes the system vulnerable. Second, holding the whole blockchain records in TEEs is not realistic due to small trusted memory (e.g. 128MB provided by Intel SGX).

Secure database designs using TEEs have been proposed to guarantee data confidentiality and integrity, while existing approaches are not suitable for maintaining a large immutable blockchain ledger. A common approach is to put the entire/partial database codes into TEEs [1]. However, this approach incurs challenges due to the TEE restrictions such as a large TCB and limitations of database functionality. Another approach is to place the whole KVS codes outside of TEEs and allow the TEE application to load/store encrypted secret data from/to KVS [2]. While it ensures confidentiality of data stored in the untrusted region, it discloses the database codes to attackers. The compromised KVS may be attacked and delete data intentionally without being noticed.

This paper proposes TEE-KV, an immutable KVS database design that allows blockchain applications running in TEEs to securely store transaction records to the whole KVS codes placed outside of TEEs. TEE-KV provides simple KVS APIs such as *Get* and *Put* for the in-TEE application to transparently communicate with the untrusted domain. To ensure immutability of data stored in the untrusted domain, TEE-KV maintains an in-TEE keyset that holds all the keys that have ever been stored in the KVS. If Get requests to the KVS do not return any value, TEE-KV can verify the correctness of the results by searching the keyset. TEE-KV is also equipped with an in-TEE cache to reduce data transfer costs between the trusted and untrusted domains.

We implemented a prototype of TEE-KV with LevelDB 1.1.8 and Intel SGX functions. Lines of code (LOC) of in-TEE domain of the prototype is only 15.1% of the LevelDB code, which leads to smaller TCB size than an all-in-TEE approach which puts the whole KVS code in the TEE. In the experiments, we integrated our TEE-KV in the framework of Coco [3] and constructed a blockchain network with three nodes. We then evaluated the throughput of Ethereum blockchain transactions in the network. The results show that the transaction throughput without in-TEE cache achieves 0.95x of the all-in-TEE approach. The results also show that 4MB in-TEE cache improves the throughput to 1.03x.

SoCC '18, October 11–13, 2018, Carlsbad, CA, USA

ACM ISBN 978-1-4503-6011-1/18/10.
https://doi.org/10.1145/3267809.3275475

REFERENCES

[1] Chia che Tsai, Donald E. Porter, and Mona Vij. 2017. Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*. USENIX Association, Santa Clara, CA, 645–658. https://www.usenix.org/conference/atc17/technical-sessions/presentation/tsai

[2] Alexey Gribov, Dhinakaran Vinayagamurthy, and Sergey Gorbunov. 2017. StealthDB: a Scalable Encrypted Database with Full SQL Query Support. *CoRR* abs/1711.02279 (2017). arXiv:1711.02279 http://arxiv.org/abs/1711.02279

[3] Microsoft. 2017. Coco Framework Whitepaper. https://github.com/Azure/coco-framework/blob/master/docs/Coco Framework whitepaper.pdf. (2017).