A Log-Structured Merge Tree-aware Message Authentication Scheme for Persistent Key-Value Stores

Igjae Kim*, J. Hyun Kim, Minu Chung,

Hyungon Moon , Sam H. Noh

UNIST, KAIST*

Persistent KVSs accommodating data

Persistent
Key Value Store

RocksDB

Redis

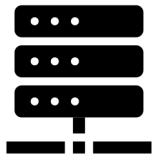
DynamoDB

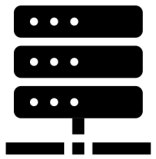
Web services

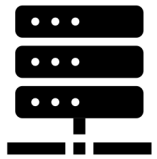
Netflix

Facebook

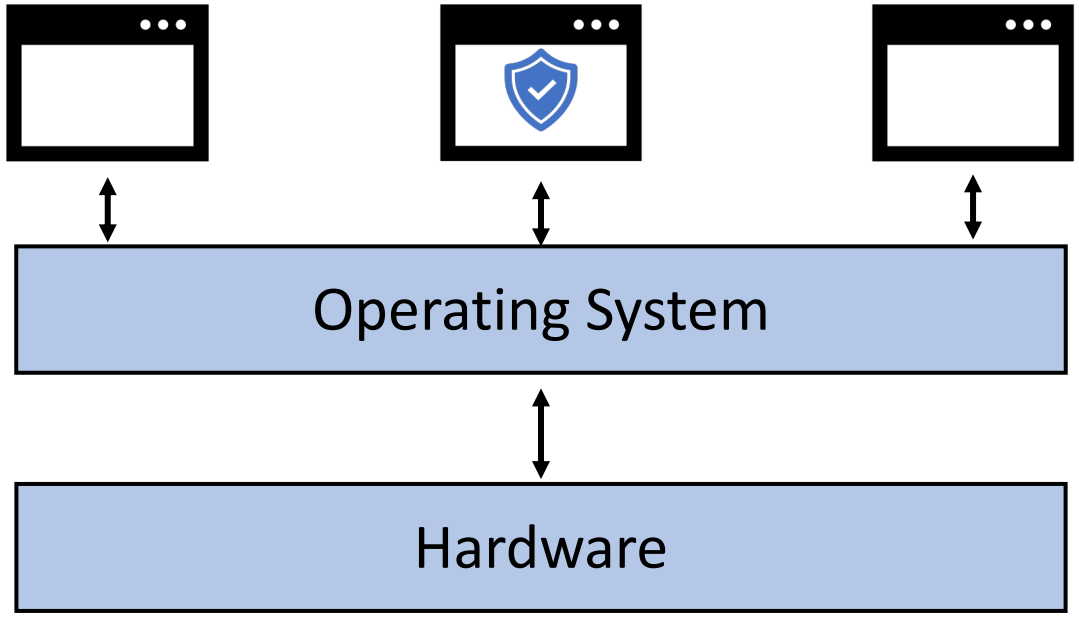
Uber







KVSs need to run in an enclave



Hardware-based confidential computing

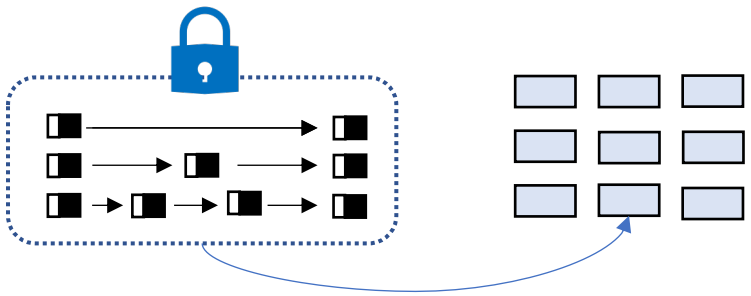
Confidentiality	Integrity	Freshness
Only authorized programs can read data	Only authorized programs can modify data	No replay of old data

Existing system: Speicher (FAST '19)

Identified and addressed three challenges

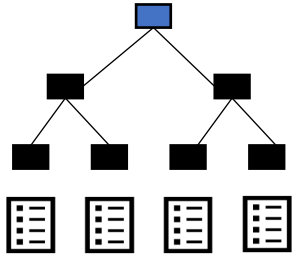
Challenges 1

- Capacity of protected memory region is limited in Intel SGX.



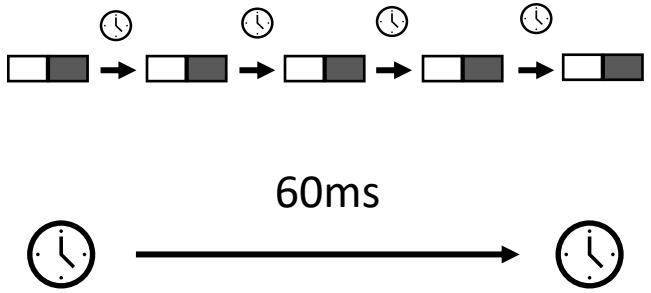
Challenges 2

- Intel SGX doesn't provides protection for data stored in storage.



Challenges 3

- Processor provided protection mechanisms are inefficient.

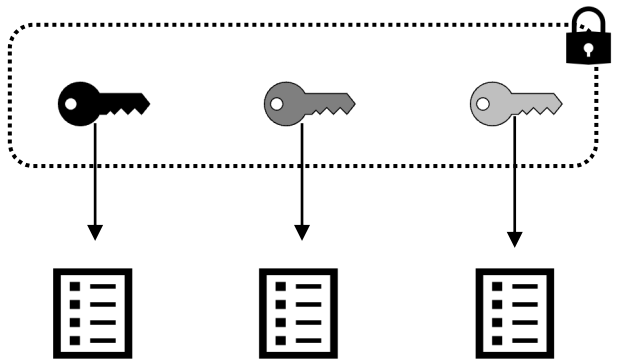


Our system: Tweezer

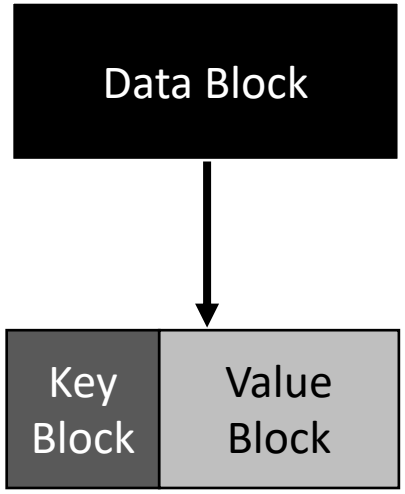
We built Tweezer on top of LSM tree efficiently alleviating these challenges.

SSTable authentication

Use Per-SSTable Key

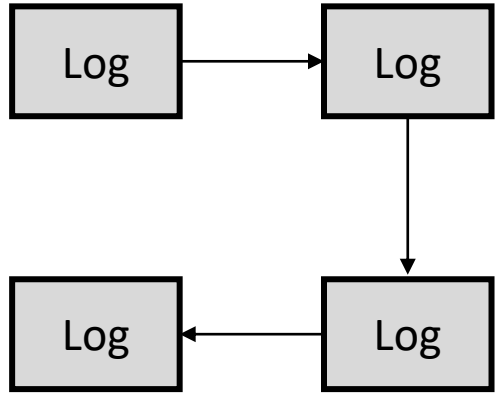


Fine-grained Authentication

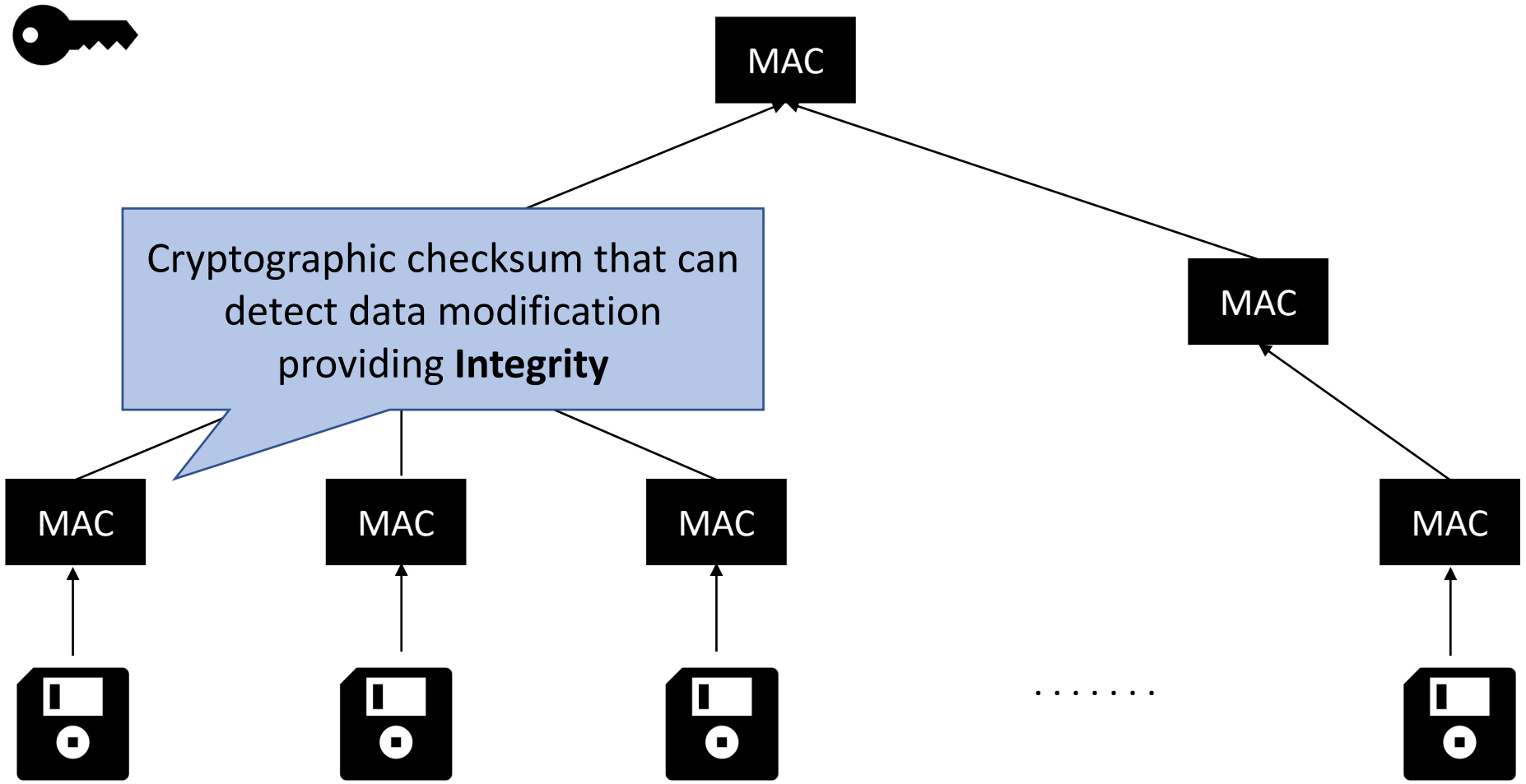


Log authentication

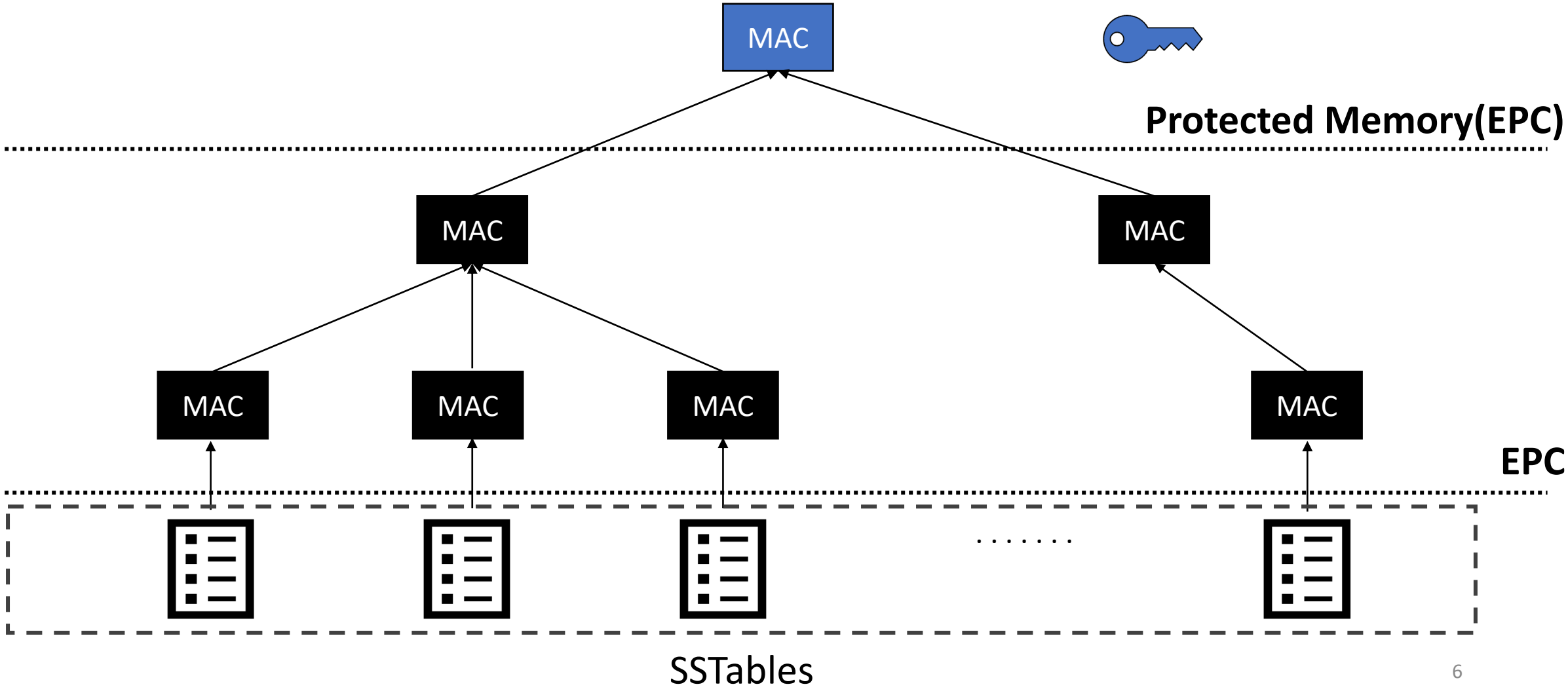
Protecting Logs with Hash Chains



Authentication with merkle tree

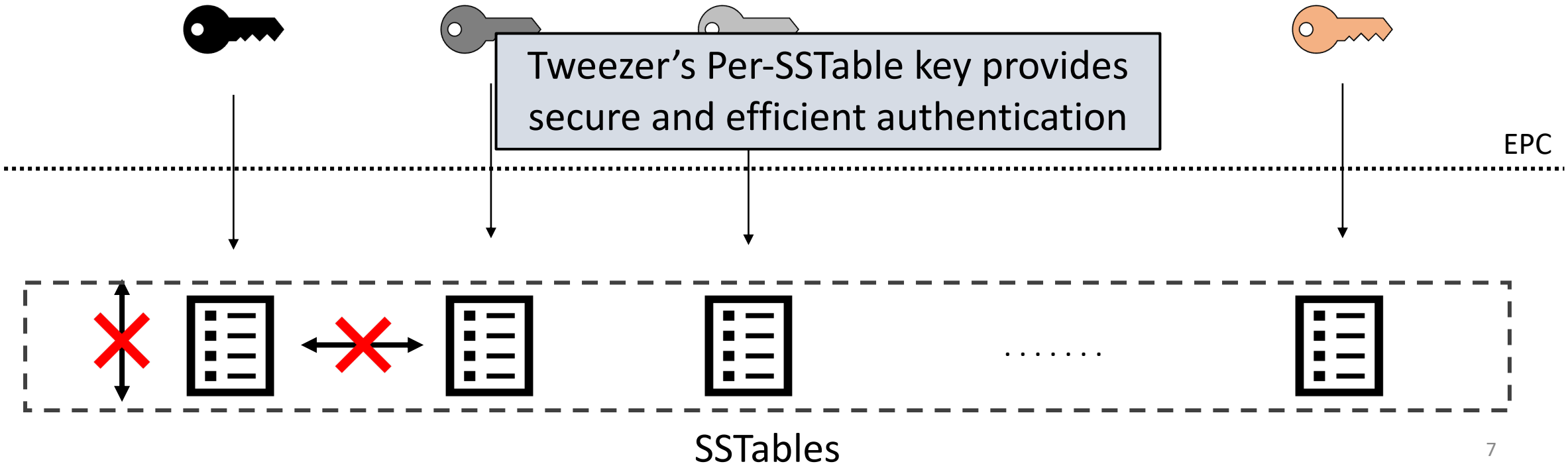


Authentication with merkle tree

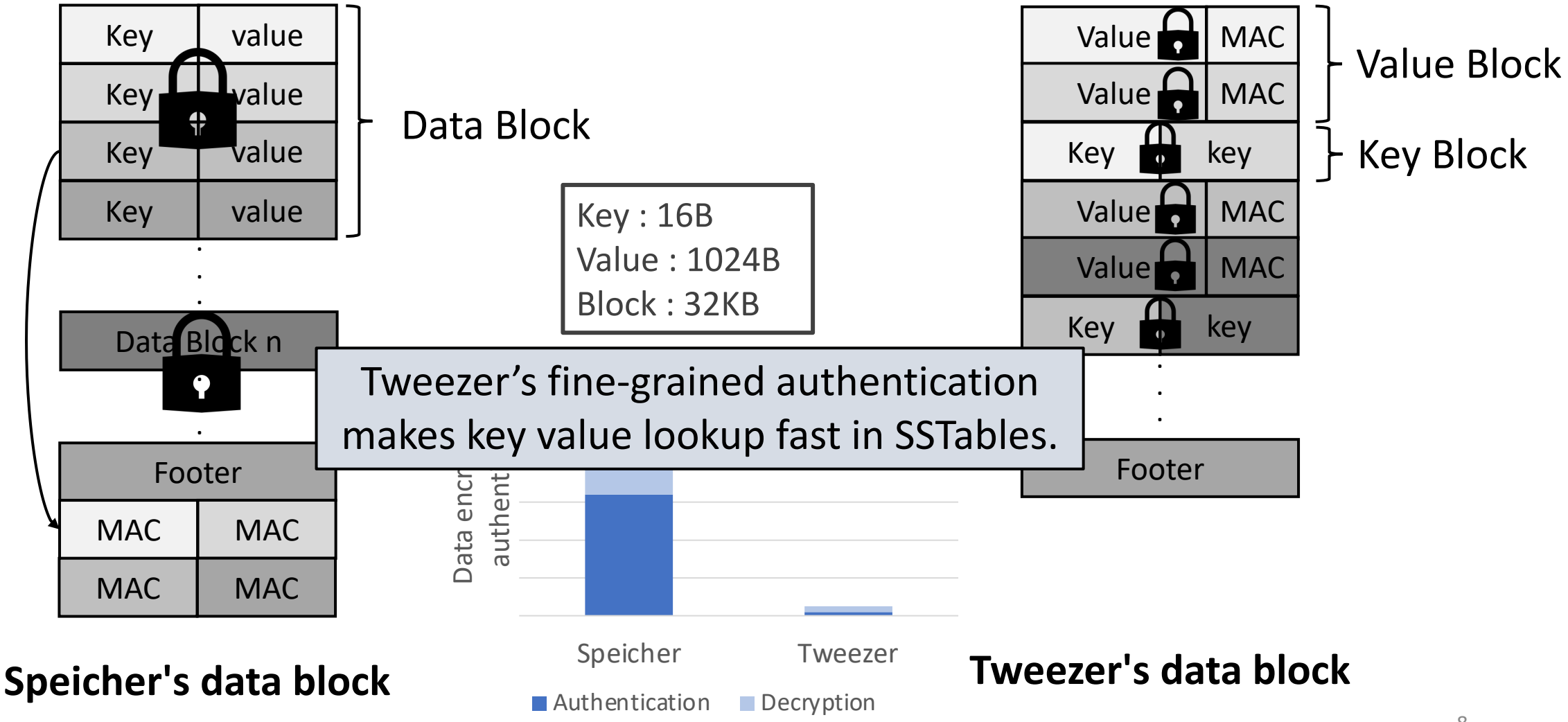


Authentication with per-SSTable Key

- Static Sorted Table(SSTable)
  - Immutable
  - Keys are sorted and unique



Fine-grained authentication

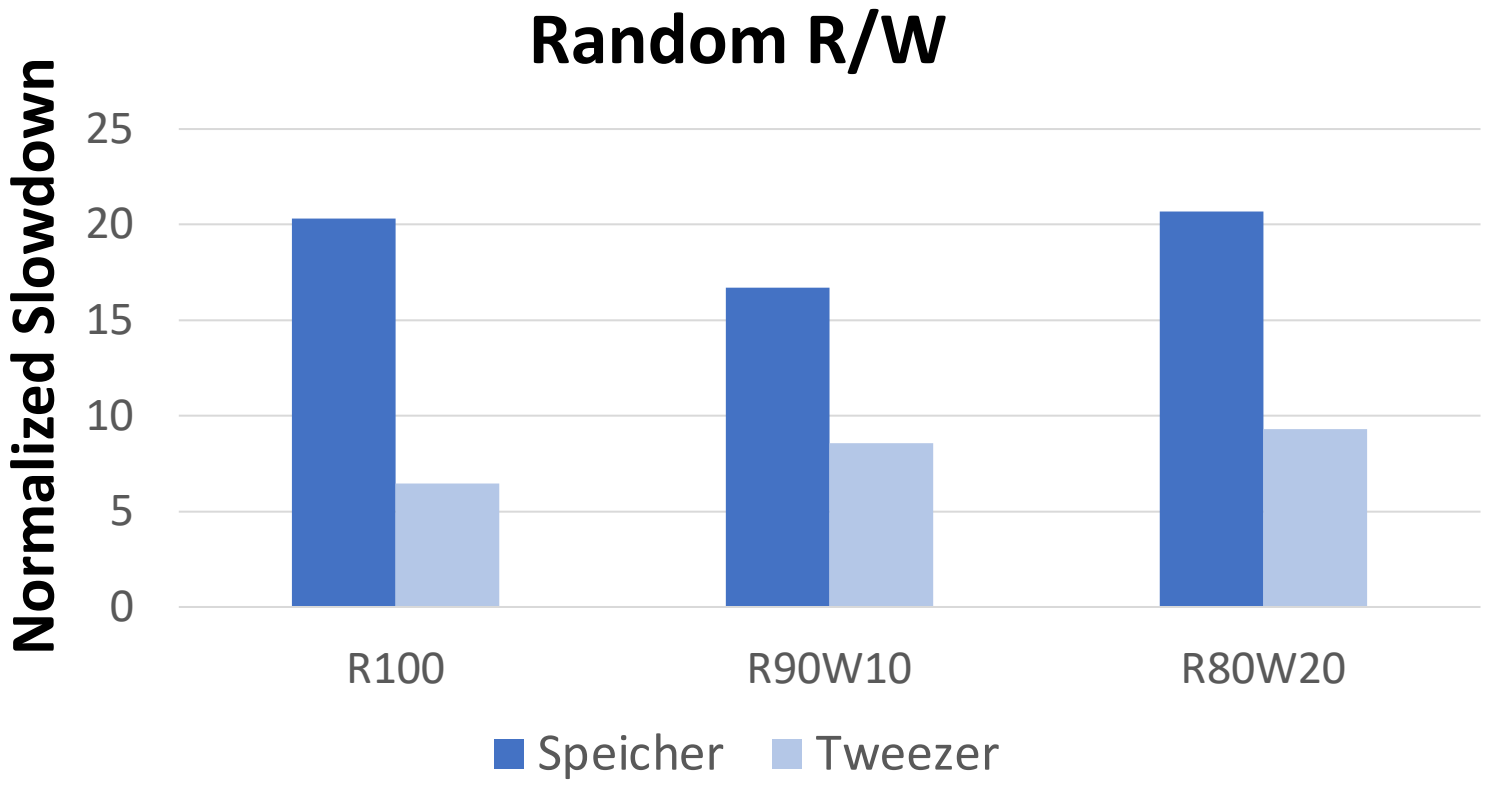


Evaluation

CPU	Intel Xeon E-2288G
EPC(Protected memory)	256MB
Main memory	64GB
OS	Ubuntu 18.04 with Linux Kernel 4.15
Crypto library	OpenSSL 1.1.1i with AES-NI

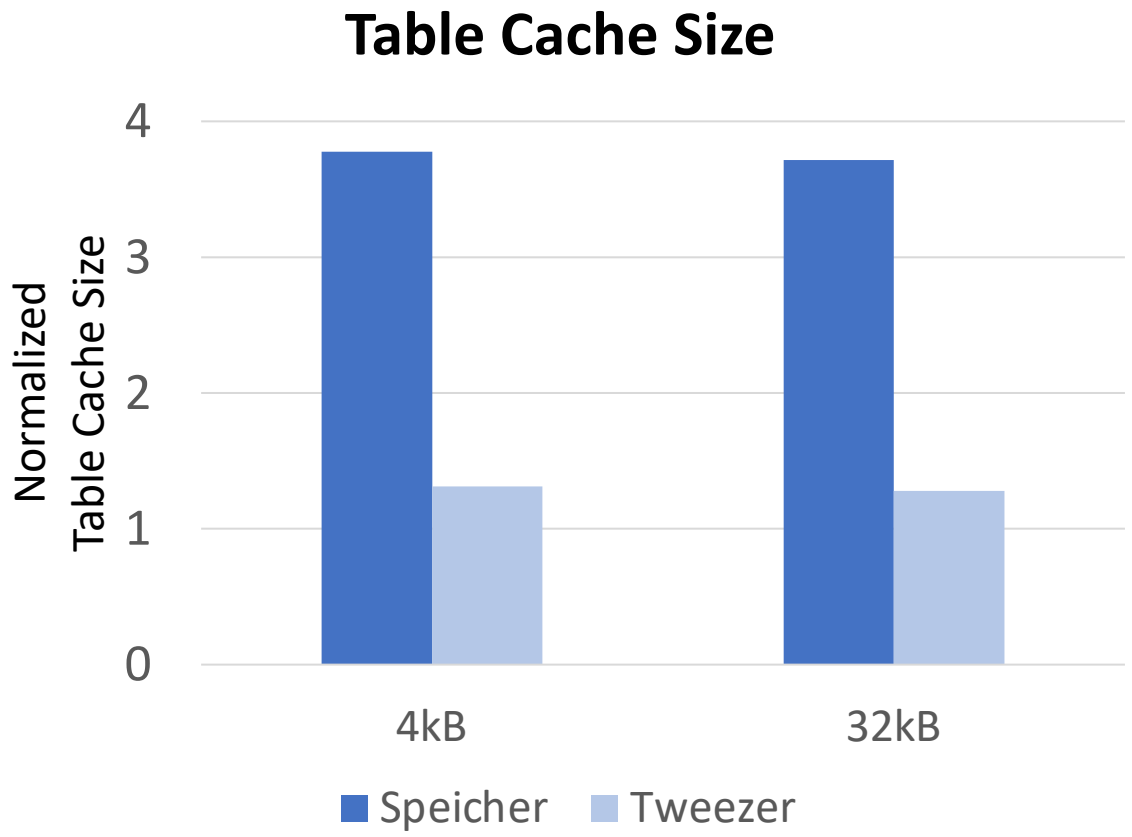
We reproduced Speicher for comparison study.

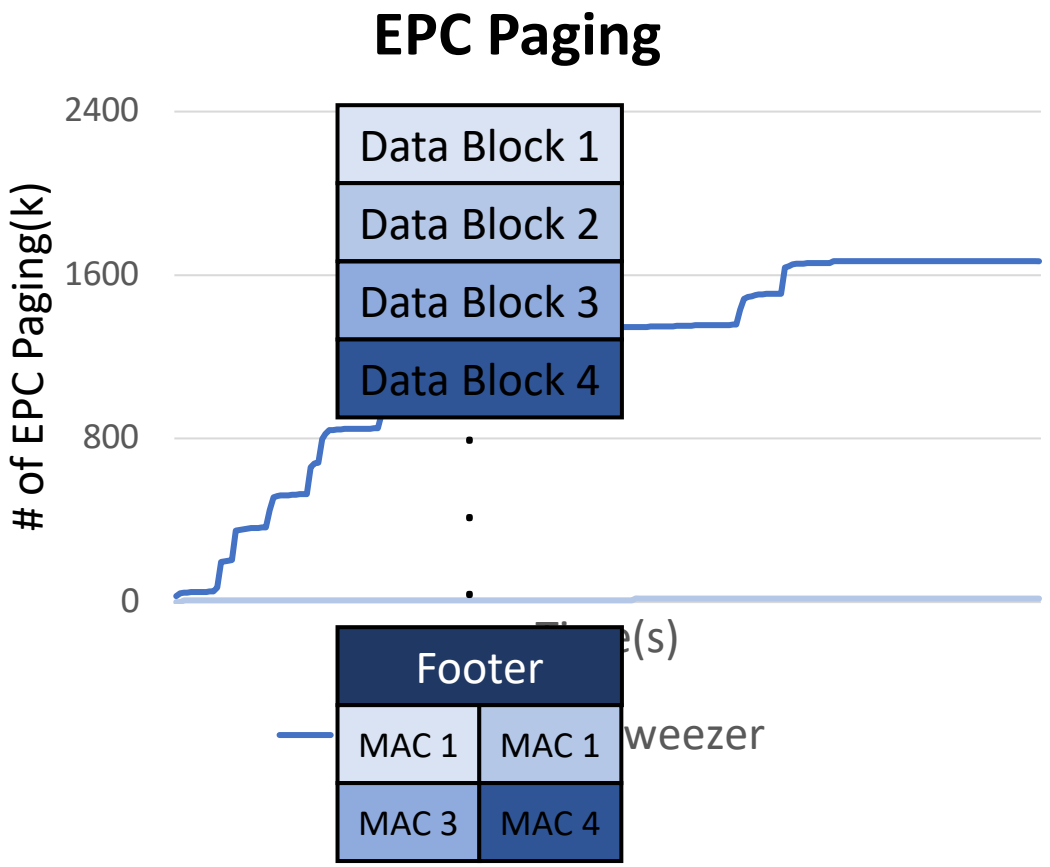
Tweezer outperforms Speicher



Tweezer outperform Speicher about
1.46 ~ 6.23x on point lookup

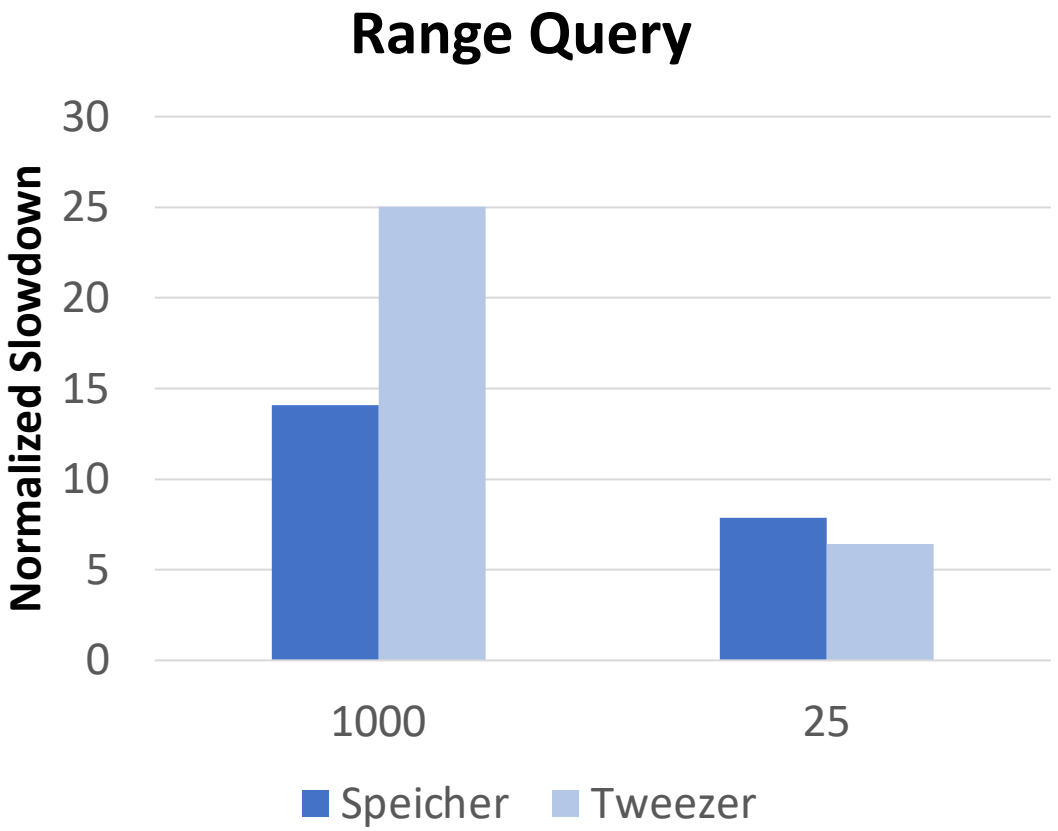
Tweezer leverage trusted memory efficiently

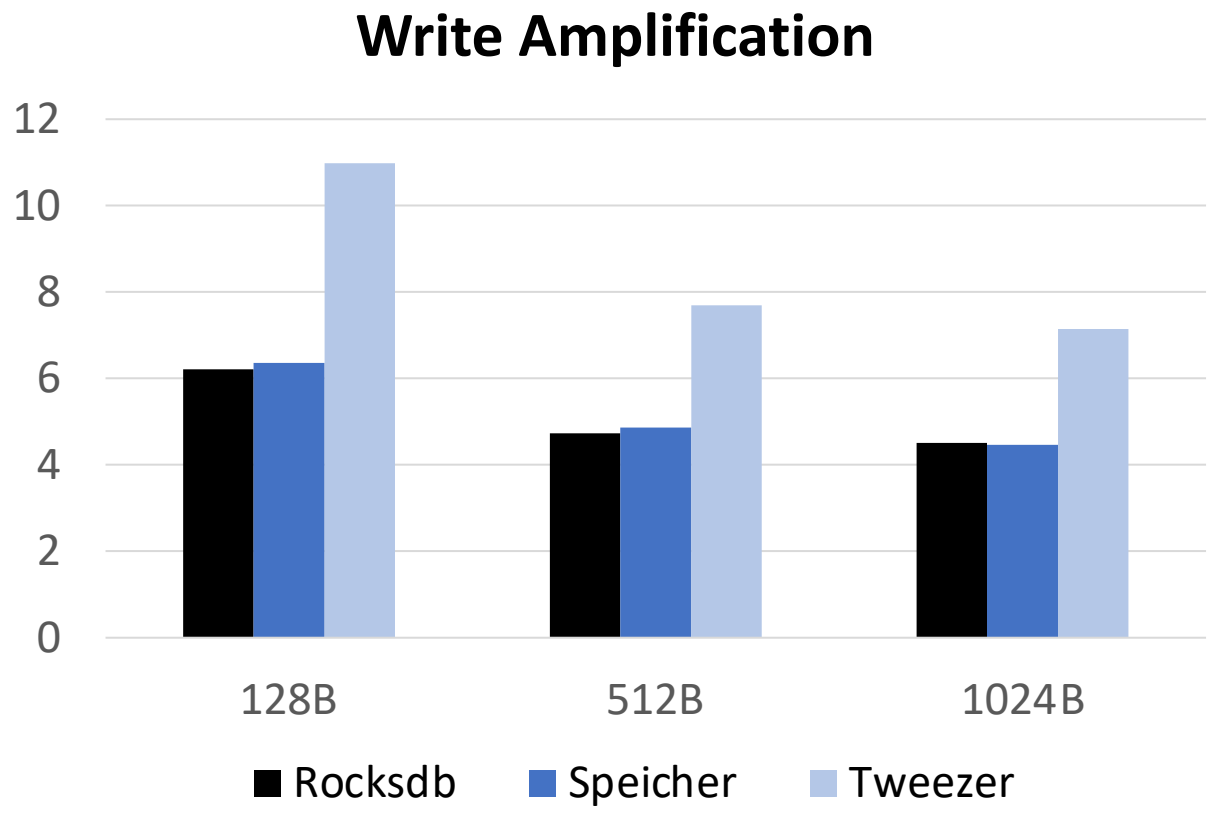




Tweezer efficiently leverages invaluable trusted memory and scales better

Drawbacks of tweezer





Range query is problematic, need further optimization

High write amplification

Conclusion

- We alleviate challenges of confidential computing efficienty leveraging characteristics of LSM tree.
  - ➢ Authentication with Per-SSTable Key
  - ➢ Fine-grained Authentication
  - ➢ Protecting Logs with Hash Chains

Thank you!