MosaicDB: An Efficient Trusted / Untrusted Memory Management For Location Data in Database

Tomoya Suzuki, Taisho Sasada, Yuzo Taenaka, Youki Kadobayashi

Nara Institute of Science and Technology (Japan)

Contact email: **suzuki.tomoya.sp9@is.naist.jp**

About me

Tomoya Suzuki

suzuki.tomoya.sp9@is.naist.jp

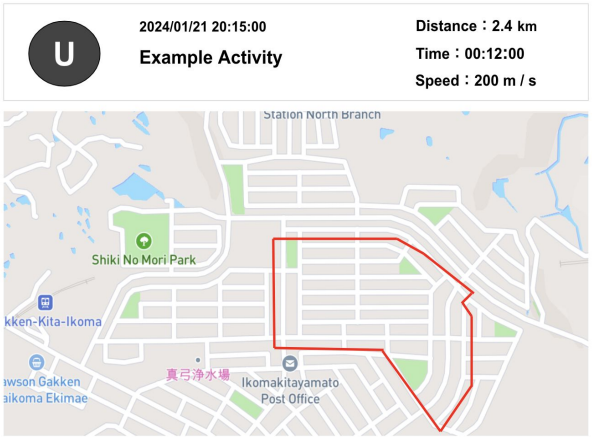
I am currently enrolled in the master's program at the Nara Institute of Science and Technology (NAIST) in Japan. My research interest lies in secure databases using a trusted hardware.

Contents

Background : Privacy issues in Fitness Tracking Social Networks

- Location data has been used for various purposes in digitized society.
- Fitness Tracking Social Networks (FTSN) collect individual location data and allow users to track outdoor activities and share their routes with other users.
- However, sharing routes raises privacy concerns, as other users can browse routes that often include sensitive waypoints, such as home or workplaces.

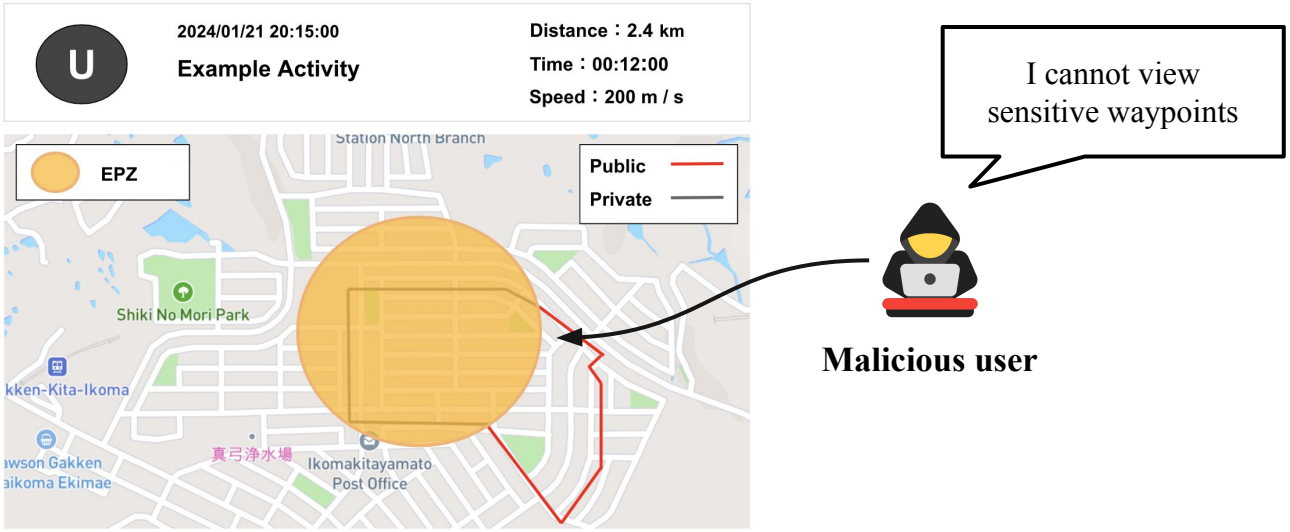
Strava [1] is the one of the most popular FTSN.



Example of sharing routes in FTSN

Background : Privacy protection with Endpoint Privacy Zones

- FTSNs enable users to designate Endpoint Privacy Zones (EPZs) to prevent privacy leakage from sharing routes. An EPZ allows users to hide some routes.
- Ongoing research [2][3] is also being conducted to implement more robust EPZs.



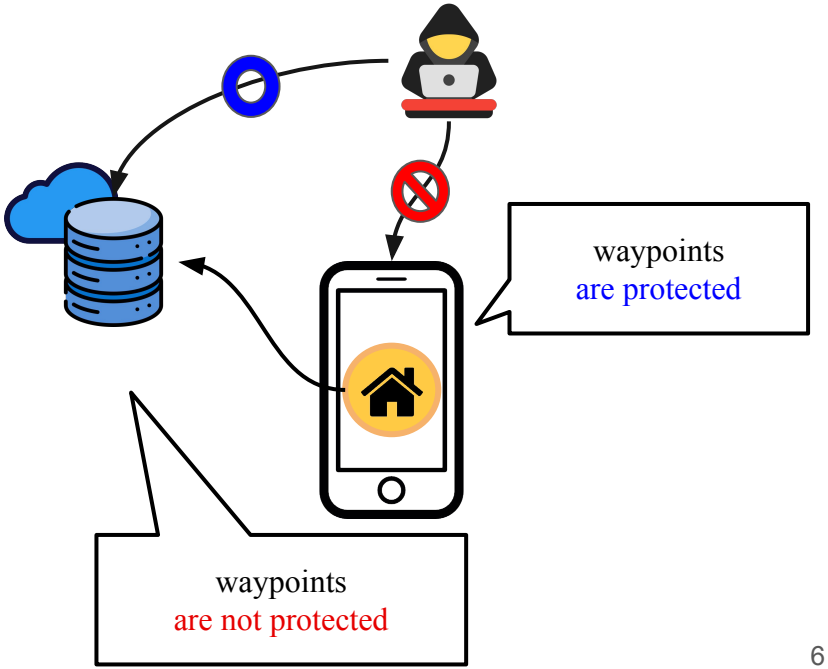
Example of EPZ in FTSN

Problem : Waypoints on a database is not protected in a cloud

- As databases are deployed in a cloud environment, waypoints including those within the EPZ in the database may be stolen by the Cloud Service Provider (CSP) with the highest privileges on the cloud system.

Problem

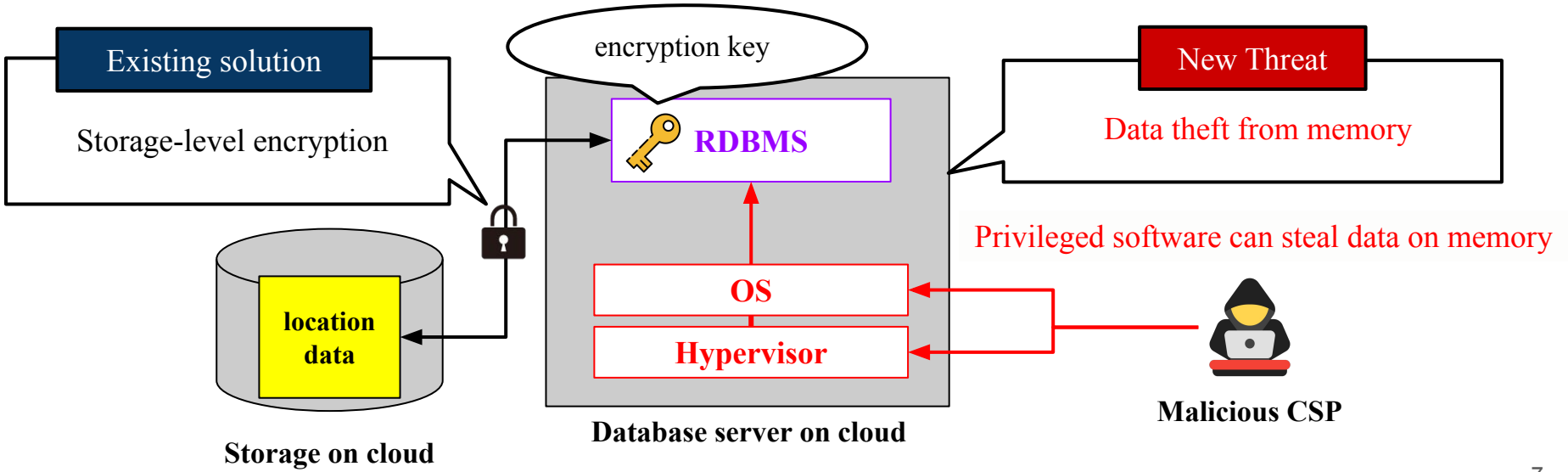
Waypoints are protected on an application. However, waypoints on a database still suffer from an exposure risk in a cloud environment.



Problem : Data theft by malicious CSPs

Malicious CSPs can steal location data (waypoints) from database directly.

- They have complete control over hypervisors and operating systems in cloud.
- They can steal all waypoints, including sensitive waypoints in the EPZs.

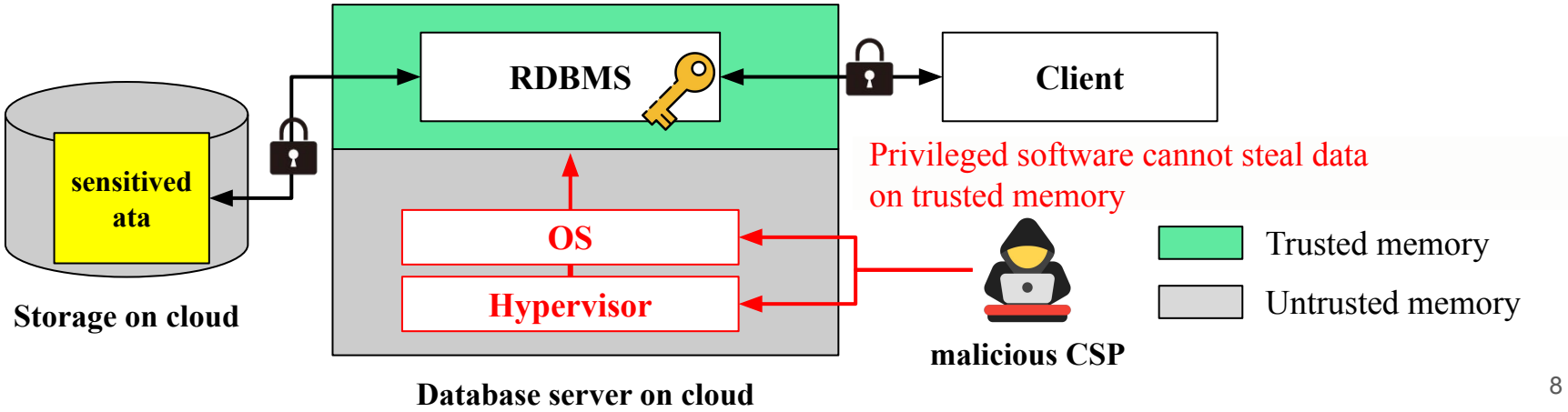


※ RDBMS = Relational Database Management System

Solution : Database with Trusted Execution Environments (TEE)

TEE can protect database memory from malicious software.

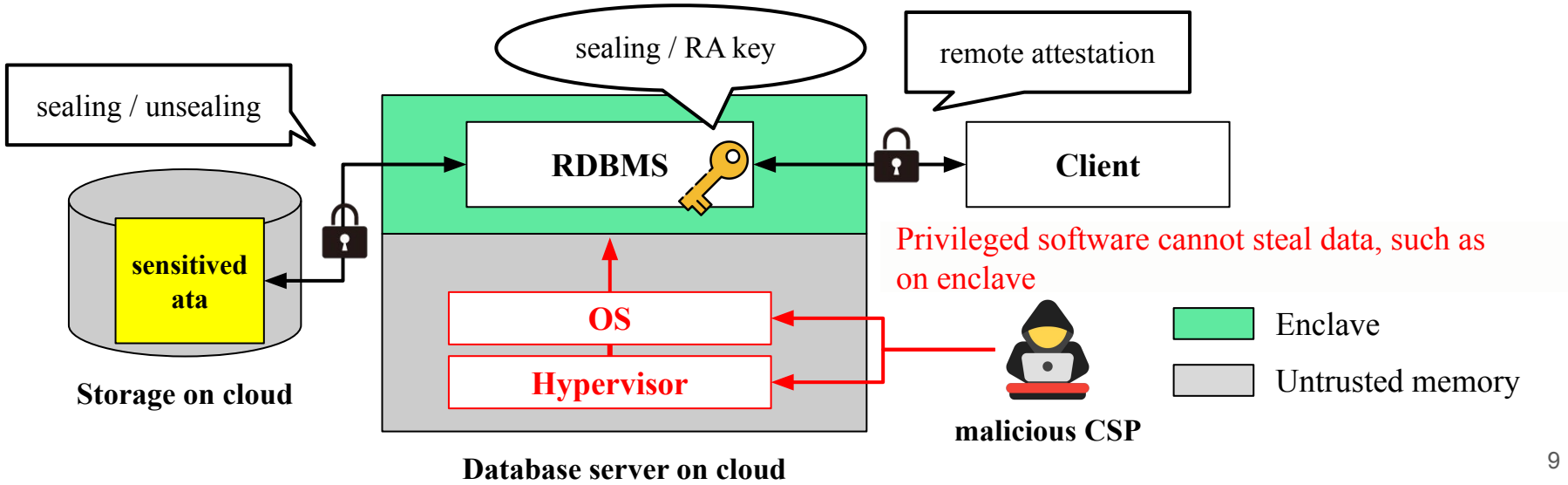
- TEE creates isolated and encrypted memory space (called trusted memory).
- Privileged software cannot read and write the data / code in trusted memory.
- Intel Software Guard Extensions (SGX) is the most widely used TEE in cloud.



Solution : Intel Software Guard Extension (SGX)

Intel SGX provides confidentiality and integrity of a program code/data with enclave

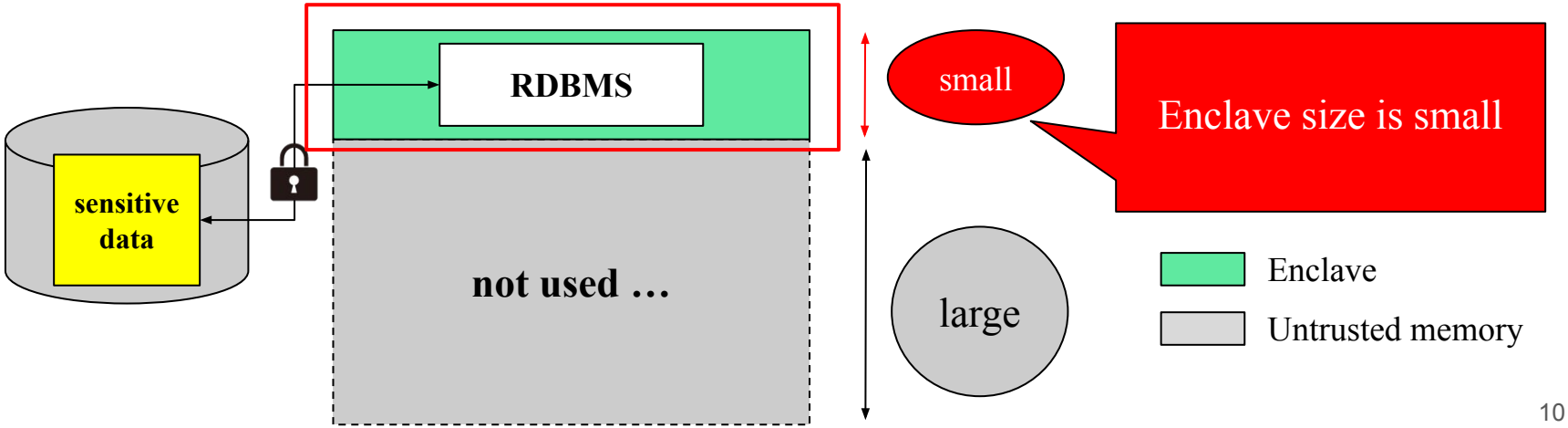
- SGX provides enclave as trusted memory space in TEE.
- SGX enable users to persist data securely with sealing / unsealing.
- SGX provides secure communication with remote client using remote attestation (RA).



Challenges in a database with Intel SGX

SGX severely limits the size of the enclave.

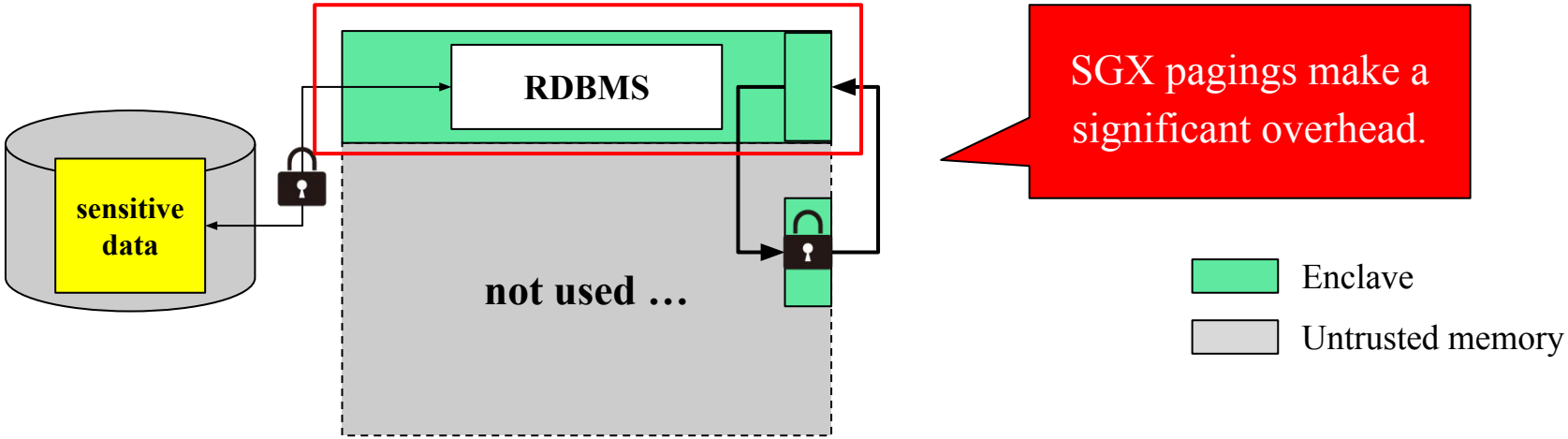
- Existing database using SGX [4][5][6][7] cannot use entire memory in a database server because they handles almost all data in the enclave.
- This design leads to enclave memory shortage and performance degradation of a database due to SGX pagings.



Challenges in a database with Intel SGX

SGX severely limits the size of the enclave.

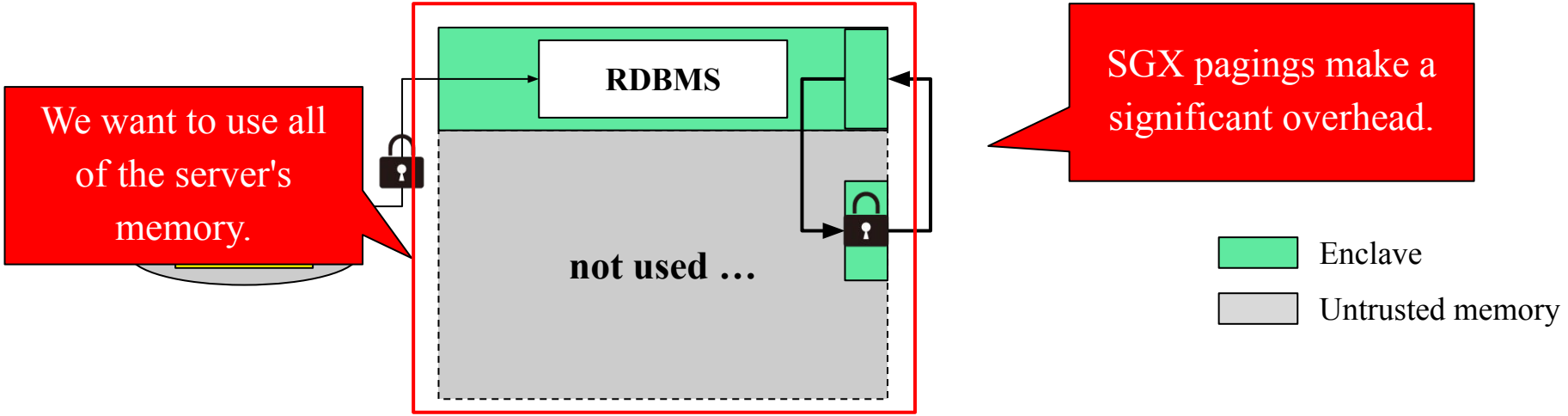
- Existing database using SGX [4][5][6][7] cannot use entire memory in a database server because they handles almost all data in the enclave.
- This design leads to enclave memory shortage and performance degradation of a database due to SGX pagings.



Challenges in a database with Intel SGX

SGX severely limits the size of the enclave.

- Existing database using SGX [4][5][6][7] cannot use entire memory in a database server because they handles almost all data in the enclave.
- This design leads to enclave memory shortage and performance degradation of a database due to SGX pagings.



A research objective and our approach

A research objective is …

Designing a database that optimally uses both enclave and untrusted memory

Our approach is …

1. Enabling query execution in both enclave and untrusted memory
2. Executing queries that handle sensitive waypoints exclusively in the enclave
3. Integrating a mechanism to identify sensitive waypoints into the database

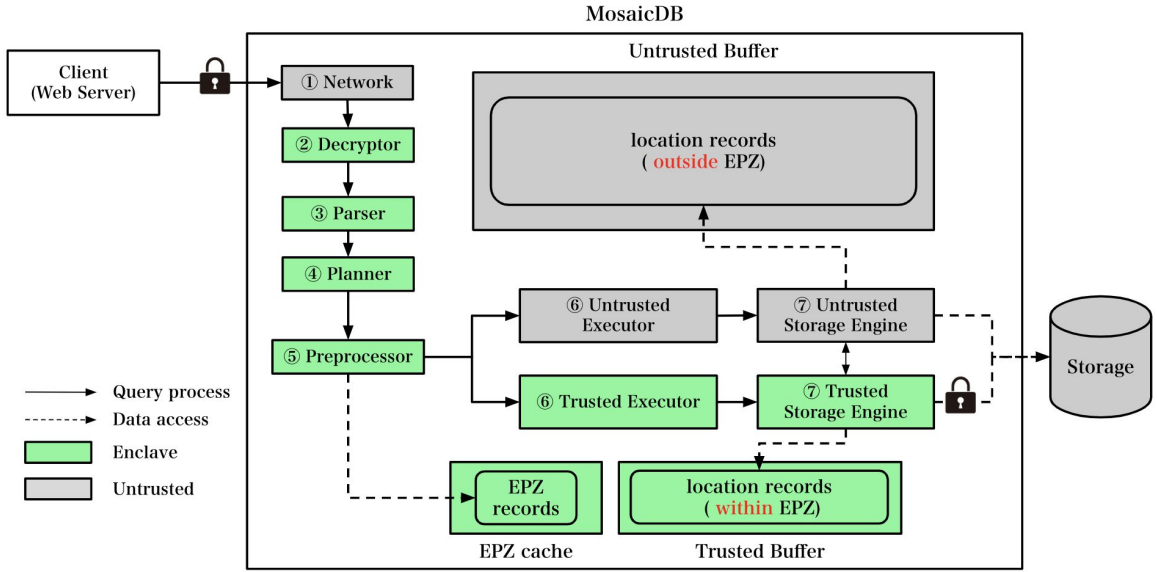
Proposed method : MosaicDB

Trusted and memory-efficient database for Location data

- MosaicDB selectively handles location data in the enclave, following the necessity of data protection in the application context.

1. MosaicDB duplicate Executor and Storage Engine.

2. MosaicDB checks whether location data is within the EPZs during INSERT operations to execute queries that handle sensitive waypoints exclusively in the enclave.



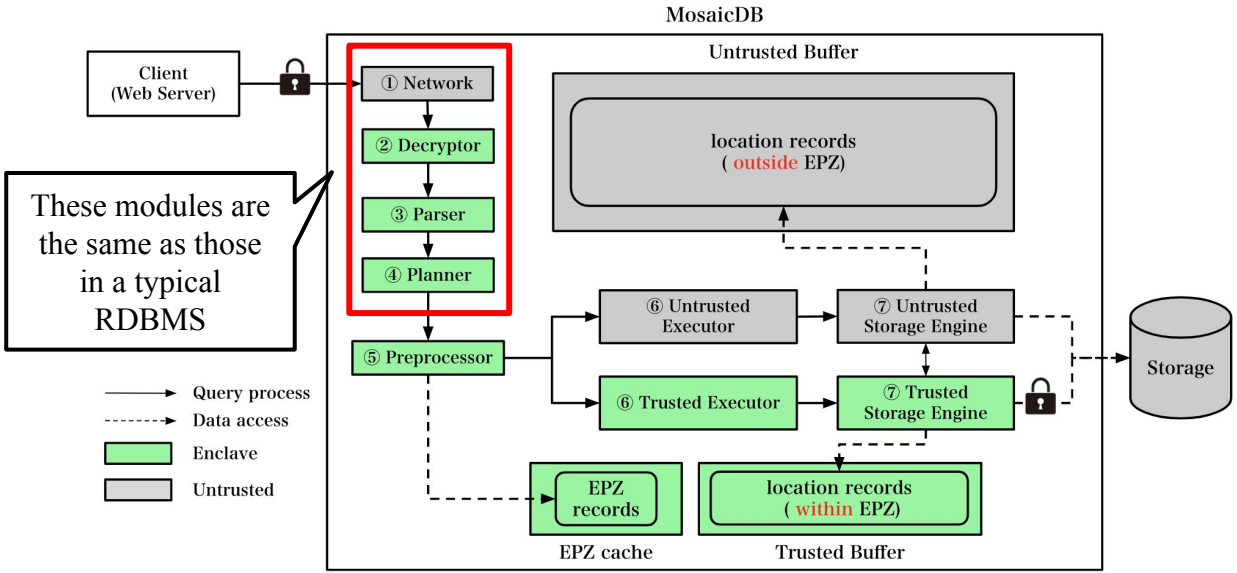
MosaicDB architecture

Proposed method : MosaicDB

MosaicDB selectively handles location data in the enclave,
following the necessity of data protection in the application context.

1. MosaicDB duplicate Executor and Storage Engine.

2. MosaicDB checks whether location data is within the EPZs during INSERT operations to execute queries that handle sensitive waypoints exclusively in the enclave.



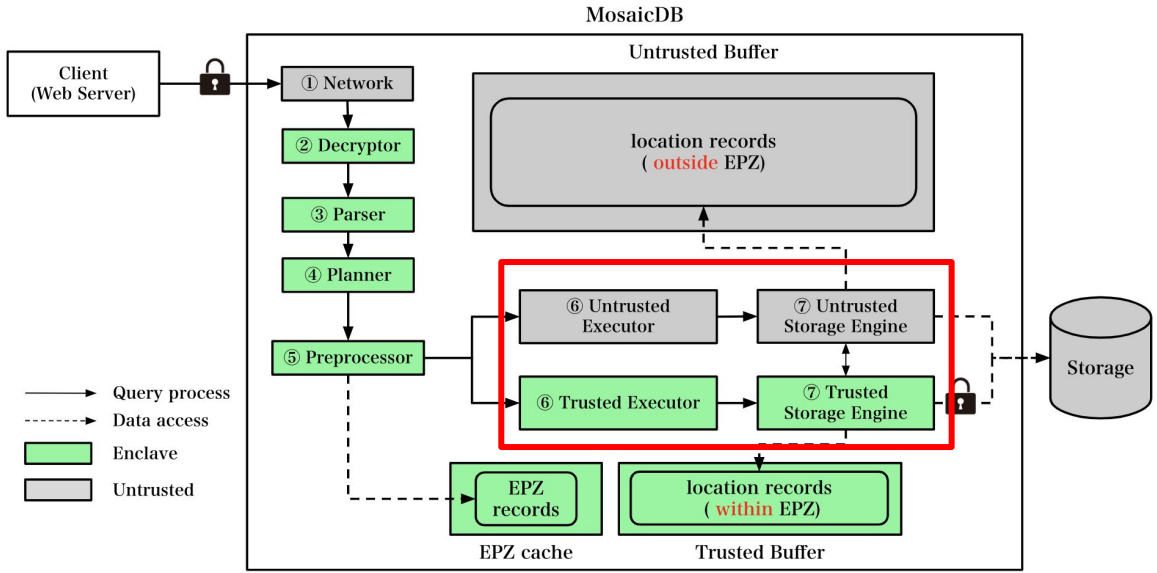
MosaicDB architecture

Enabling query execution in both enclave and untrusted memory

The executor and storage engine within the enclave handle queries involving sensitive location data, whereas those in untrusted memory execute queries related to non-sensitive location data.

1. MosaicDB duplicate Executor and Storage Engine.

2. MosaicDB checks whether location data is within the EPZs during INSERT operations to execute queries that handle sensitive waypoints exclusively in the enclave.



MosaicDB architecture

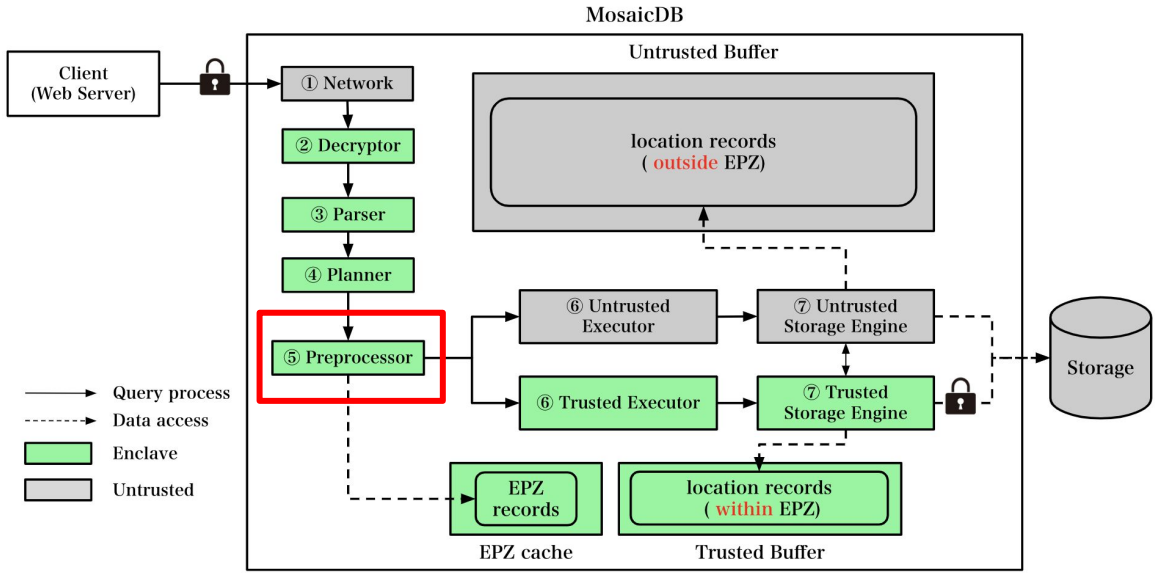
Identifying sensitive location data

Preprocessor checks whether the location is contained within the EPZ.

- TEE creates isolated and encrypted memory space (called trusted memory).

1. MosaicDB duplicate Executor and Storage Engine.

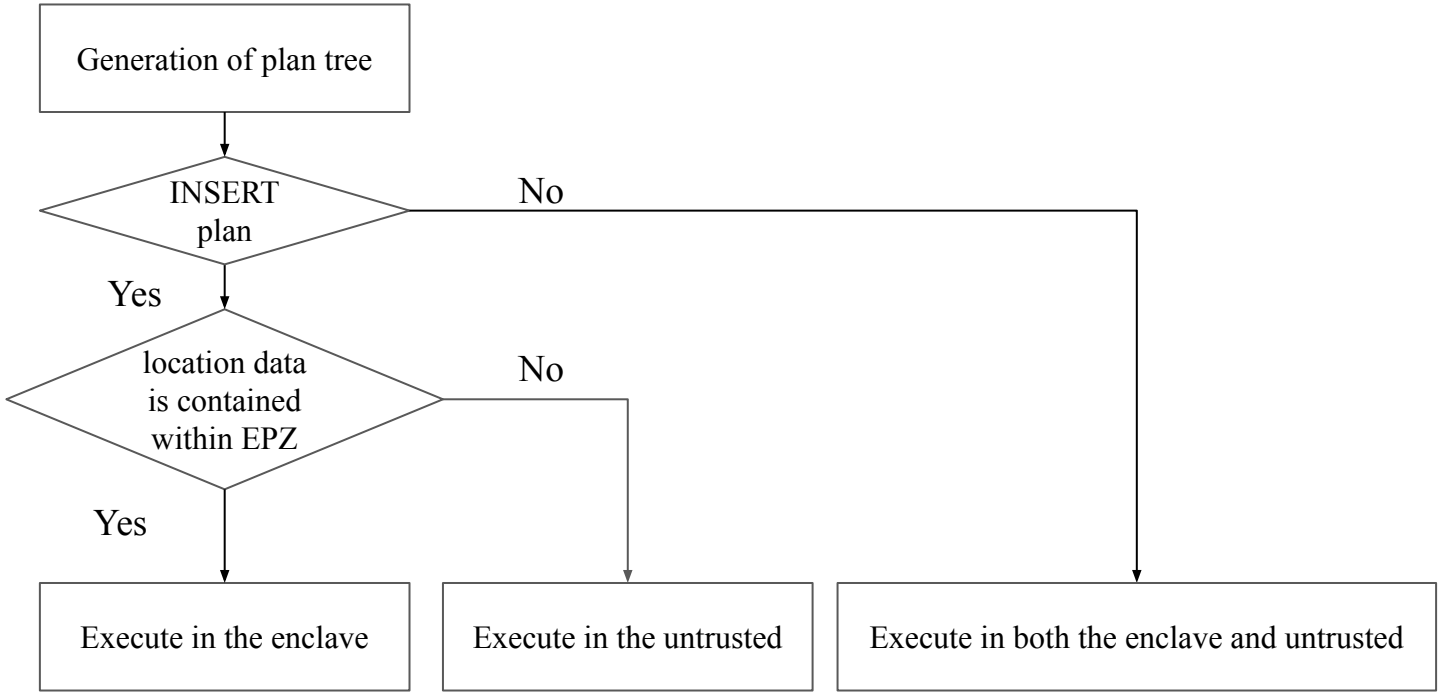
2. MosaicDB checks whether location data is within the EPZs during INSERT operations to execute queries that handle sensitive waypoints exclusively in the enclave.



MosaicDB architecture

Identifying sensitive location data

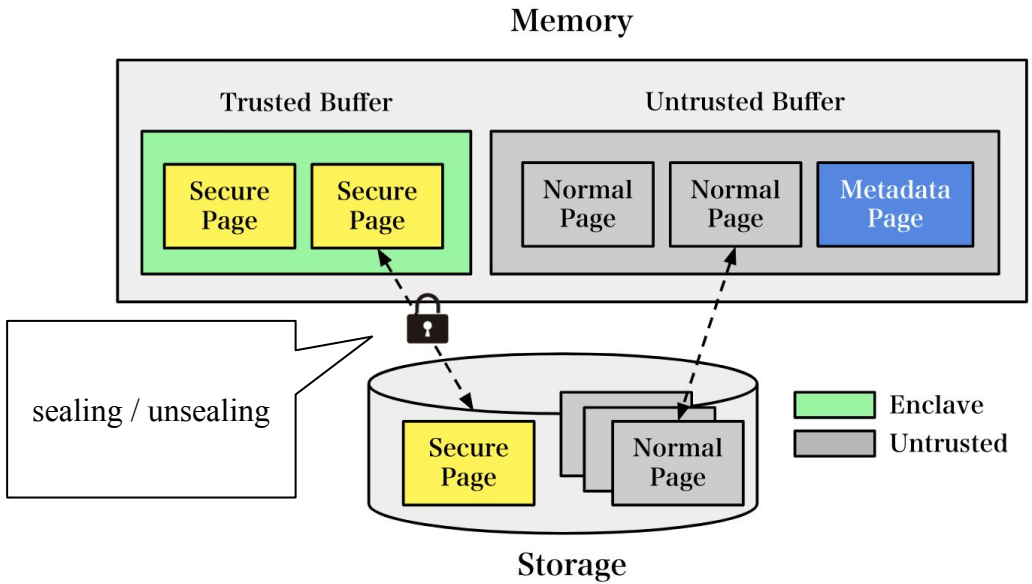
Preprocessor checks whether the location is contained within the EPZ.



Page management in MosaicDB

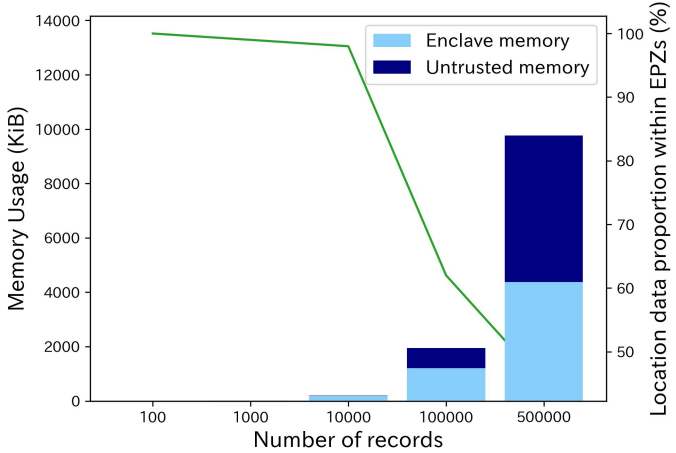
Secure pages contain sensitive location data.

Secure pages will be encrypted with SGX sealing before persistence.



Experiments : Estimation of memory usage

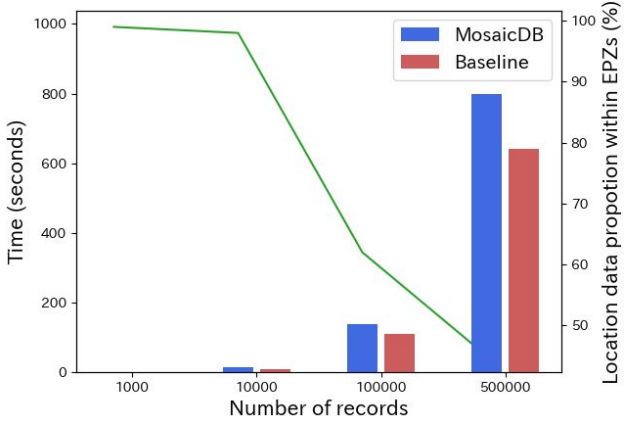
- Results shows that both the enclave and untrusted memory are used when the number of records exceeds 100,000.
- Decrease in the proportion of location data within the EPZs led to improved memory utilization efficiency.



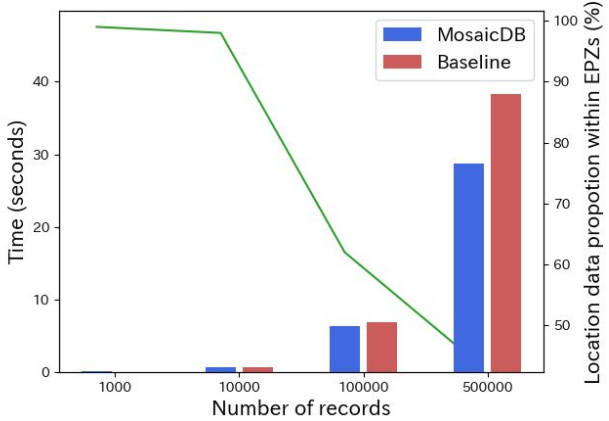
※ memory usage = record size * data size

Experiments : Execution time (INSERT・SELECT)

- Results shows that MosaicDB increases an overhead by 1.2 to 1.6 times compared to the baseline in INSERT query
- Results shows that MosaicDB can reduce SELECT query execution time by up to 25% compared to the baseline



Execution time of INSERT



Execution time of SELECT

Conclusion and future work

Conclusion

- We proposed MosaicDB, a memory-efficient and trusted database that manages location data using both the enclave and untrusted memory in SGX
- MosaicDB improved memory utilization efficiency
- MosaicDB achieved a 25% reduction in execution time for selection queries

Future work

- We will integrate existing transaction mechanism like ARIES into MosaicDB.
- We will measure enclave memory load by monitoring SGX pagings.

References

[1] https://www.strava.com/
[2] W. U. Hassan, S. Hussain, and A. Bates, “Analysis of privacy protections in fitness tracking social networks -or- you can run, but can you hide?,” in 27th USENIX Security Symposium, pp. 497–512, USENIX Association, Aug. 2018
[3] K. Dhondt, V. Le Pochat, A. Voulimeneas, W. Joosen, and S. Volckaert, “A run a day won’t keep the hacker away: Inference attacks on endpoint privacy zones in fitness tracking social networks,” in Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 801–814, Association for Computing Machinery, 2022
[4] Y. Wang et al., “Cryptsqlite: Sqlite with high data security,” IEEE Transactions on Computers, vol. 69, no. 5, pp. 666–678, 2019
[5] C. Priebe, K. Vaswani, and M. Costa, “Enclavedb: A secure database using sgx,” in 2018 IEEE Symposium on Security and Privacy, pp. 264–278, 2018
[6] A. Gribov, D. Vinayagamurthy, and S. Gorbunov, “Stealthdb: a scalable encrypted database with full sql query support,” in Proceedings on Privacy Enhancing Technologies Symposium, pp. 370–388, 2019
[7] M. Yoshimura, T. Sasada, Y. Taenaka, and Y. Kadobayashi, “Memory efficient data-protection for database utilizing secure/unsecured area of intel sgx,” in DBKDA 2023, The Fifteenth International Conference on Advances in Databases, Knowledge, and Data Applications, pp. 38–43, 2023

Attribution

All icons used in this slide were from https://www.flaticon.com/.