Mimer SQL utilizing Trustonic's Trusted Execution Environment - White Paper

02/06/2020

Introduction

A key component to protect the confidentiality of information is encryption. Encryption is the process of transforming data into an unintelligible scrambled format using algorithms, while decryption unscrambles the data making it readable again. Both encryption and decryption require a key. This ensures that only authorized parties, who have access to the key, can read the data. Encryption is frequently used to protect data at rest that is stored on computers, databases, or storage devices.

The advantage of database encryption is that data stored in the database cannot be read by unauthorized individuals that do not have the key. However, since the strength of the encryption relies on the key remaining secret, it is important to consider where the key should be stored. A poor choice of key storage can render the encryption worthless since anyone who has access to the key can read the data.

Trustonic's Trusted Execution Environment

Kinibi is Trustonic's [1] Trusted Execution Environment operating system which can execute Trusted Applications on a device using ARM TrustZone technology. TrustZone is implemented by hardware manufactures and provides a high degree of hardware isolation between "trusted" and "untrusted" code. Kinibi uses TrustZone to separate the Rich Execution Environment (REE), i.e. Android, from the Trusted Execution Environment (TEE) where Trusted Applications run. The TEE is used to protect and isolate sensitive assets, such as sensitive code or information. Kinibi has been integrated in over one billion devices of smartphones, tablets, laptops, and IoT devices.

Mimer SQL Key Storage Solution

Mimer SQL [2] is using AES-encryption to encrypt data stored on disk. The data is decrypted when queried and loaded into the system memory or cache. To solve the problem of database encryption key storage, Mimer SQL is leveraging Trustonic's Trusted Execution Environment (TEE), Kinibi, to provide a hardware isolated secure storage for the encryption key. All code execution that uses the key, such as encryption and decryption, is executed by Mimer's Trusted Application running in the TEE. This way, the key is protected from potentially malicious applications or code running in the Rich Execution Environment. Mimer's Trusted Application is also completely isolated from other Trusted Applications that may exist in the TEE. The architecture of the solution can be seen in figure 1.

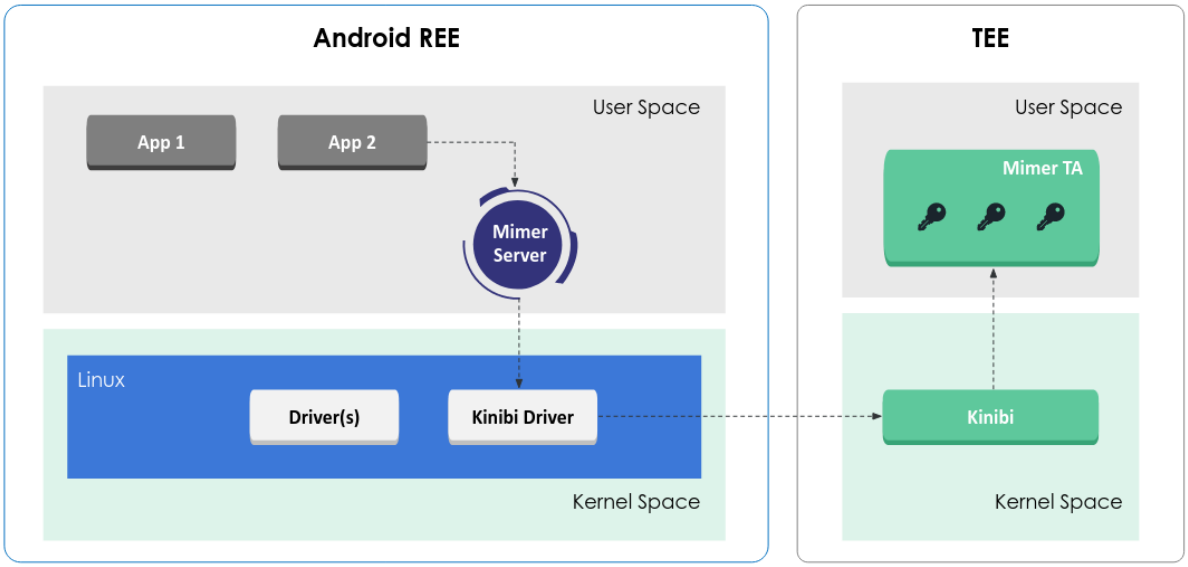

Figure 1: Architecture overview of Mimer SQL's integration with Kinibi.

With the help of Mimer's Trusted Application, the encryption key is protected throughout its lifetime. The key is generated inside the "secure world" of the TEE and then stored inside the TEE's secure storage, which can only be accessed by Mimer's Trusted Application running inside the TEE. This means that the encryption key is protected at all times from potentially malicious applications running in the Rich Execution Environment operating system. The key being protected from unauthorized users results in the encrypted information in the database files remaining protected as well, as it cannot be decrypted without the key. This is a great advantage in case of a device being lost or stolen since the information in Mimer SQL's encrypted database files cannot be read even with physical access to the device.

This solution is very beneficial for security, however, it is also important to consider how this solution affects performance. Since encryption and decryption operations are executed in the Trusted Execution Environment, this means that the Mimer server has to make calls to another operating system.

To examine the performance impact performing encryption and decryption on Mimer's Trusted Application, several test were performed on a HiKey-board. The tests are described in table 1. The tests all included inserting 5000 rows into a table using different methods.

Table 1: Descriptions of the different tests performed.

Test1	Inserting 5000 rows without prepared statements
Test2	Inserting 5000 rows using prepared statements with implicit transaction handling
Test3	Inserting 5000 rows using batched statements and auto-commit.
Test4	Inserting 5000 rows using batched statements and explicit transaction handling. Transaction size: 100

The results of the tests can be seen in figure 2.

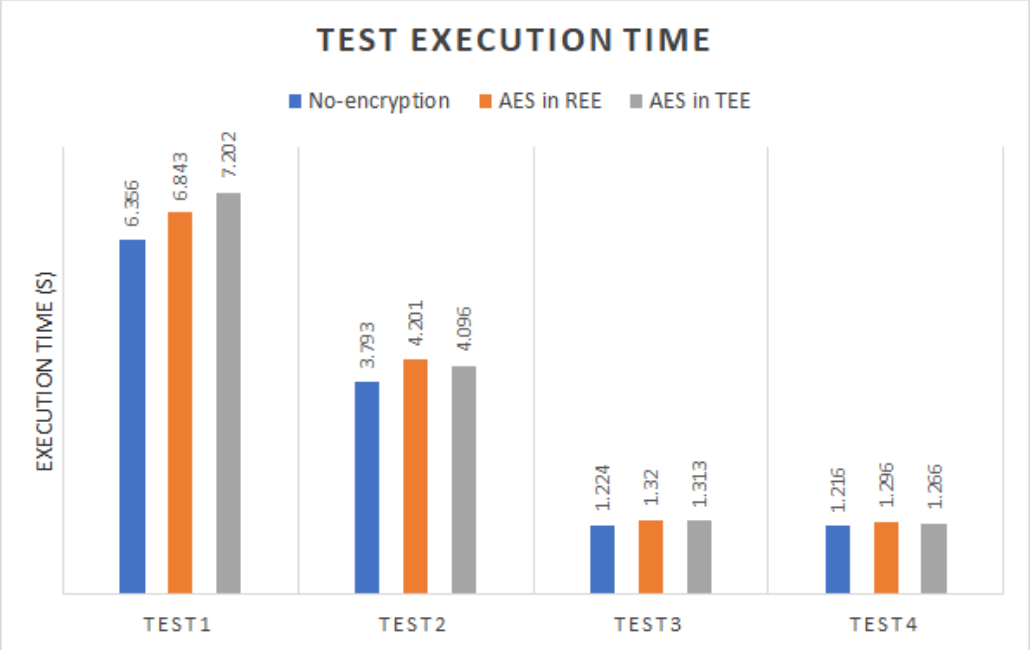

Figure 2: Performance tests results.

The results showed an overall minimal impact on the execution-time when utilizing the Mimer Trusted Application for encryption and decryption. Only in test 1, the performance difference is more noticeable. In tests 2, 3, and 4, prepared and batched statements were used and this resulted in minimal time differences. The Mimer server opens a session with the Mimer Trusted Application when starting, and then keeps this session open waiting for commands to encrypt or decrypt data. This also has a positive impact on the time-performance.

Conclusion

Mimer SQL is using a Trusted Application running inside Trustonic's Trusted Execution Environment. The database encryption key is stored in a hardware isolated secure storage, well protected from potentially malicious code running in the rich execution environment. This solution has several advantages - the key is protected but also easily accessible for the Mimer server to use when needed. The security is increased substantially with only a small loss in performance.

References

[1] *Trustonic - Enabling trust through simpler, stronger security*, `https://www.trustonic.com/`.

[2] *Home - Mimer*, `https://www.mimer.com/`.