CYBERNETICA

Public

Sharemind HI for Web Services - White Paper

Document ID: D-16-194

Version 1.0
13.02.2024

Table of Contents

1. Introduction

1.1. Purpose

Sensitive information like business secrets and personal data is widespread in all kinds of information systems. Business owners depend on the protection of their business secrets against data breaches and misuse, and personal data is heavily regulated especially since the inception of GDPR. Data breaches or data misuse could give away a crucial edge of a business to competitors, or result in severe fines from data protection agencies. Such sensitive data is present in many domains, including health and banking sectors.

In a classical architecture, as shown in Figure 1, users share sensitive and non-sensitive data with a (web) service. The service persists the data in a database, and an attacker or compromised internal account can access the sensitive data. In cloud environments, an erroneous configuration can open a database to the whole world.

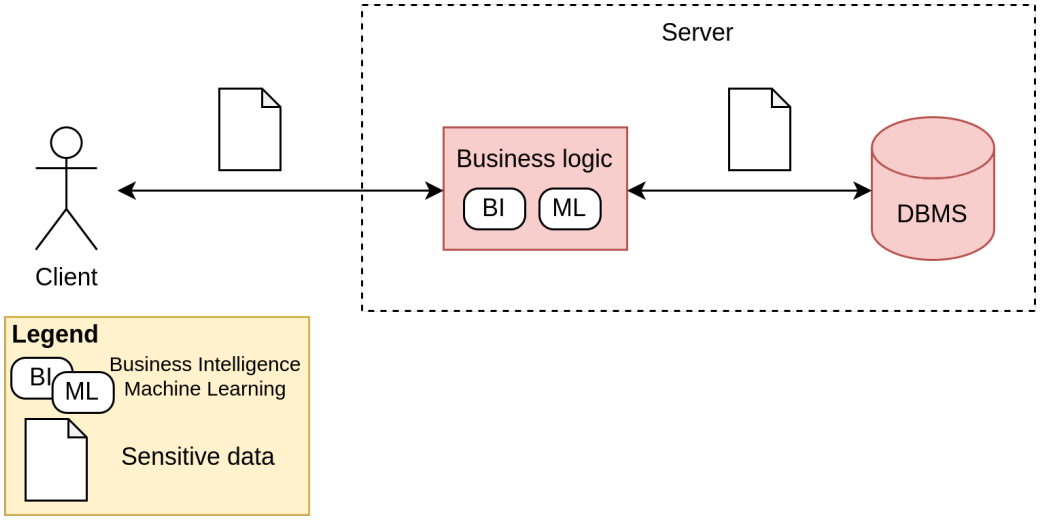

Figure 1. An unprotected web service. Users connect to the web gateway which contains the business logic, and data is persisted in a database. The business logic and DBMS have access to sensitive data. A data breach will expose sensitive data.

In this paper we introduce how you can use Sharemind HI (Hardware Isolation) to protect sensitive data from attackers and comply with GDPR while keeping your value chains intact. Sharemind HI is a platform to securely process data of any number of stakeholders, using a hardware based *Trusted Execution Environment*. Configuration and permission management in Sharemind HI adhere to the Four Eyes Principle.

Sharemind HI integrates with your existing solution and disrupts existing workflows as little as possible. Sharemind HI encrypts the sensitive data and controls who can access what data. Still, the sensitive data remains accessible to your value chains including Business Intelligence, Data Analytics and Machine Learning. The full data analysis power of Sharemind HI is at your fingertips to access all the value from the sensitive data. Sharemind HI just makes it easy to do it securely.

We have worked out a large knowledge base and catalog of documents to help you comply with GDPR regulations when using Sharemind HI.

1.2. Users

Sharemind HI for web services is designed for the providers of web services who want to protect sensitive data, to keep business secrets safe and comply with GDPR. Service providers can give strong assurances to data producers, analysts or third parties about the data only being used as agreed upon.

1.3. Technology

The Sharemind HI White Paper gave an overview over Intel® Software Guard Extensions (SGX) which is used in Sharemind HI. Intel® SGX provides a Trusted Execution Environment (TEE) which helps to protect data in use. It is available in modern Intel® server processors. For more details, please refer to the Sharemind HI White Paper.

2. Sharemind HI for Web Services

2.1. Concepts

Figure 1 shows a simplified version of a web service. Users connect to a web gateway which contains the business logic, and all their sensitive and regular data is stored in a database. The connection between the user and web gateway is usually protected through TLS, but the server sees the sensitive data in clear. Any future attacker will see the sensitive data in clear. And the user does not have any assurance about what their sensitive data will *really* be used for.

With Sharemind HI you will be able to both protect sensitive data from data breaches, *and* give users assurance about how you use their data. This is possible due to the flexibility of Sharemind HI, which acts as a key management system (KMS) and a universal computation platform with fine grained access controls. Users can remotely attest the software which runs inside of the Sharemind HI Server and upload encrypted data if they trust the software. Even a malicious operating system, which hosts the Sharemind HI Server, is incapable to alter the software or access the encrypted data.

Figure 2 shows how an existing system is extended with Sharemind HI. Sharemind HI is added as a sidecar to your existing business logic. It encrypts all sensitive data and opens the data only for users who have the necessary permissions, without ever making the decrypted sensitive data accessible to the business logic or database itself. If you have data processing workloads which access the sensitive data, like Business Intelligence and Machine Learning, then these need to be moved into the controlled environment of Sharemind HI. All the workloads which do not touch the sensitive data can stay where they are. We call this architecture the *Partially Encrypted Database* architecture, or short *PEDB*. A patent is pending for the PEDB architecture.

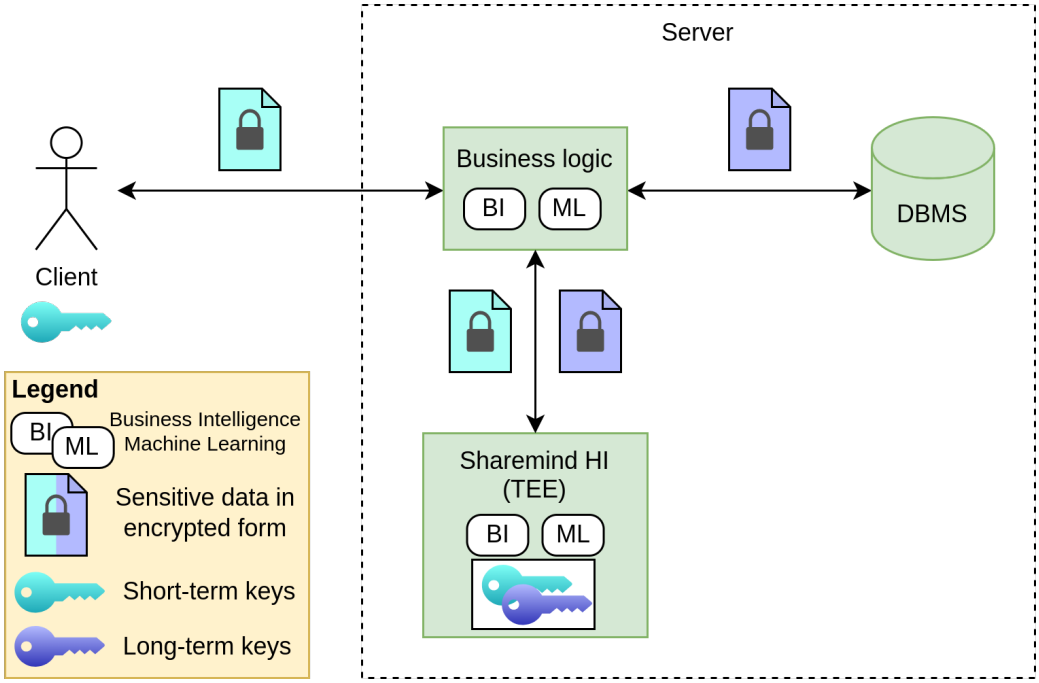

Figure 2. A web service where sensitive data is protected by Sharemind HI. Clients send and receive data which is encrypted with short term keys. The business logic stores data in the database which is encrypted with long term keys only known to Sharemind HI. A data breach will not expose any sensitive data.

The overall idea is as follows: Clients use a short-term upload key which they share securely with Sharemind HI. When they want to upload data, they encrypt the sensitive data with the upload key and send it to the business logic which forwards it to Sharemind HI. Sharemind HI decrypts the data with the short-term upload key and re-encrypts the data with a long-term storage key. *This* version of the encrypted data will be stored in the database. The short-term and long-term keys are protected with the help of Intel® SGX and will not leave Sharemind HI.
At this point, Sharemind HI can run your data processing workloads on the protected data and derive valuable information. The use of Intel® SGX allows Sharemind HI to process the data securely, such that even a person with root access to the server is incapable to inspect the protected data. Processed data may no longer need protection and Sharemind HI enables you to access such data in clear. However, when the results of your data processing workloads are still sensitive, like machine learning models, Sharemind HI can continue to protect them.
Users can access protected data by sharing a short-term download key with Sharemind HI. Sharemind HI has access to both the short-term and long-term keys, and can thus decrypt the protected data from the database and re-encrypt it with the download key of the client. A flexible access control mechanism grants users only access to the data which they are allowed to see.

3. Detailed Description

In this section we will refine the high level overview, discuss at what points of your existing infrastructure you need to integrate with Sharemind HI, and explain the workflow of common operations when using this solution architecture with Sharemind HI.

3.1. Component Overview

Figure 3 refines Figure 2. Most of your components need modifications to make use of Sharemind HI:

Client app

The client app needs to communicate with the Sharemind HI Server using a Sharemind HI client library. The messages can be exchanged by adding a new function in your existing API. The client app stores short-term keys locally and uses them to encrypt and decrypt sensitive data before sending them to the web service.

Sharemind HI client library

This library is required to communicate securely with the Sharemind HI Server. It can be used in web apps, mobile apps and Linux applications.

Web Service

The web service, or gateway, forwards client requests to the Sharemind HI Server and manages the invocation of tasks in the Sharemind HI Server, e.g. for re-encryption or machine learning application. Therefore it uses a Sharemind HI client library, too. It fetches all the data from the database for the Sharemind HI Server to process a given request, and forwards results to the database or client.

Periodic tasks

Some tasks can be done periodically in the background. If you need functionality like report generation, machine learning, etc, then the respective functionality in the Sharemind HI Server is invoked by a Sharemind HI client library, and the periodic task needs to provide all the necessary data from the DB to Sharemind HI, as was the case for the web service.

Database

Columns with sensitive data need to be modified to contain binary values instead. The sensitive data is encrypted and gets an additional encryption header with meta information.

Sharemind HI Server

A stand-alone executable which exposes a gRPC interface. It uses the local file system for persistent storage of its data. The functionality like re-encryption or report generation is written in the C++ language using the Sharemind HI SDK.

Remote Monitoring Service

An optional service for enhanced security monitoring of the Sharemind HI Server, which is controlled by another party. You can extend this service to suite your needs.

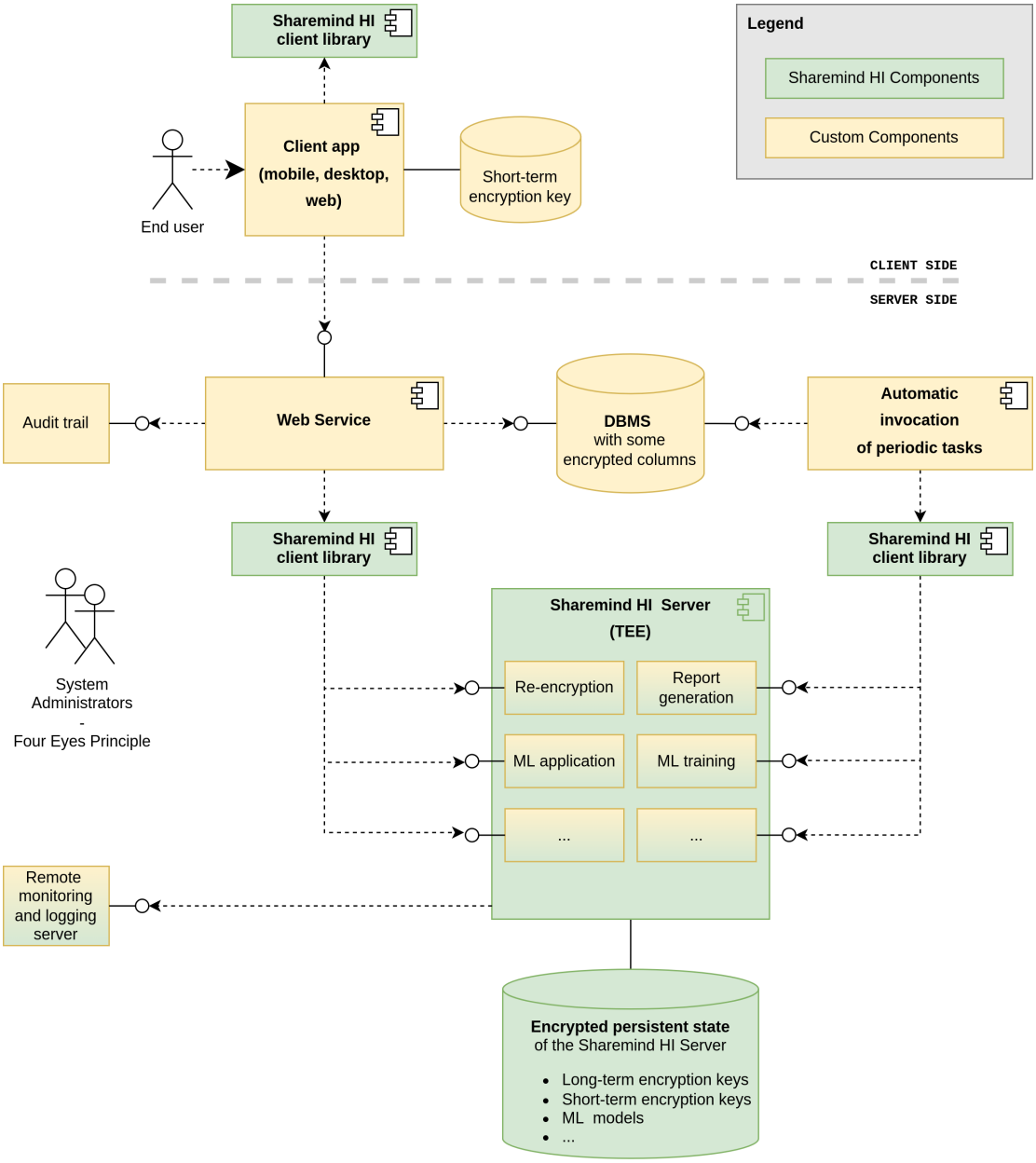

Figure 3. A more detailed view of the involved components when integrating Sharemind HI with an existing web service.

3.2. Data Protection Mechanism

Sensitive data in database fields is protected using Authenticated Encryption with Associated Data (AEAD encryption). This is used to add integrity protection, such that the sensitive data in a specific field cannot be swapped with similar fields from other rows, columns or tables, as well as adding additional restrictions for access control. Additional metadata needs to be stored for each field.

Sensitive data is encrypted at the source, and only decrypted within the TEE of Sharemind HI or in an authorized user device. Other components only see encrypted data.

3.3. User Management & Access Control

The user management and access control is managed by at least two individuals (following the four eyes principle), like an administrator from the IT department and an administrator from the QA department. Ideally they are from two different departments to reduce the risk of collusion. The admins can add users to the system by signing the users' X509 certificates[1] and signing a set of permissions. This procedure allows to easily recover lost key material of users, but needs the four eyes principle to prevent an administrator from maliciously issuing permissions and allowing users or themselves to leak sensitive data.

The signed permissions are verified by the Sharemind HI Server within the TEE whenever a user requests access to sensitive data.

3.4. Workflows

There are two common types of workloads in this scenario:

Small requests which are immediately answered
: When reading data from the database, including precomputed reports. Or when a machine learning model is used to predict something.

Larger analytics
: These can be performed periodically in the background, and their results are valid for a longer period of time.

Sharemind HI is flexible and allows additional, more custom analytics to be performed in addition to the ones shown in this example. In the following we concentrate on the above use cases to showcase the overall workflow.

3.4.1. Re-encryption

Re-encryption is necessary when users insert or read sensitive data. Users use short-term encryption keys to protect sensitive data, but the sensitive data in the database is protected with long-term encryption keys. A user creates a short-term key e.g. when they open the client app, and share the key with the Sharemind HI Server. The long-term keys are generated by the Sharemind HI Server itself and never moved out of the TEE. Hence, the Sharemind HI Server knows all short-term keys and long-term keys and can translate between the two sides.

Access control allows to decide on a field level whether the user is allowed to access the data or not.

3.4.2. Machine Learning

Machine learning consists of two steps: Training the model, and applying the model on user inputs for prediction. Model training is a resource intensive, time consuming process and may require a large input from the database. Hence it is periodically invoked, and the resulting model is used to perform all the predictions until a newer version of the model will be trained.

The model is protected by the Sharemind HI Server, too, as it maybe is a business secret itself.

Model application is often a lot faster and can be done in real time. The results of the model application can be encrypted directly with the short-term keys, or first stored in the database using long-term keys and then read through the re-encryption workflow.

3.4.3. Report Generation

Reports are similar to machine learning training. They take a longer time to compute, but the content is relevant for a longer time period. Hence the report generation is invoked periodically, and new reports replace old reports. A generated report is stored in the database using long-term keys, and can be read later through the re-encryption workflow.

3.5. Development

There is an demo project which showcases the use of Sharemind HI in the context of web services. The workflows are shown from end-to-end, and developers can use it as a recipe to start the specialised solution for their use cases.

Task enclaves for Sharemind HI are written in the C++ programming language. For common tasks, as the above described workflows, helper functions are provided which abstract away the cryptographic details. This allows the developer to focus on algorithm details and access control policies.

The Sharemind HI client library is available for the C++ and TypeScript languages. The C++ version works on Linux, iOS and Android. The demo application showcases the C++ library, and how to perform encryption and decryption of fields with sensitive data.

The server side can make use of the same Sharemind HI client libraries. The database and the web service can serve multiple tenants. However, a single Sharemind HI Server should only serve a single tenant for stronger security boundaries.

4. Document History

- Version 1.0: Initial version.

[1] Sharemind HI uses an EC p256v1 asymmetric key pair. The public key is stored in a certificate which is used for authentication in the Sharemind HI Server. This key pair could be stored locally on the user device, or protected with a password and stored in a central place.