安永数据库加密系统

使用手册

（Intel SGX 虚拟机映像版本）

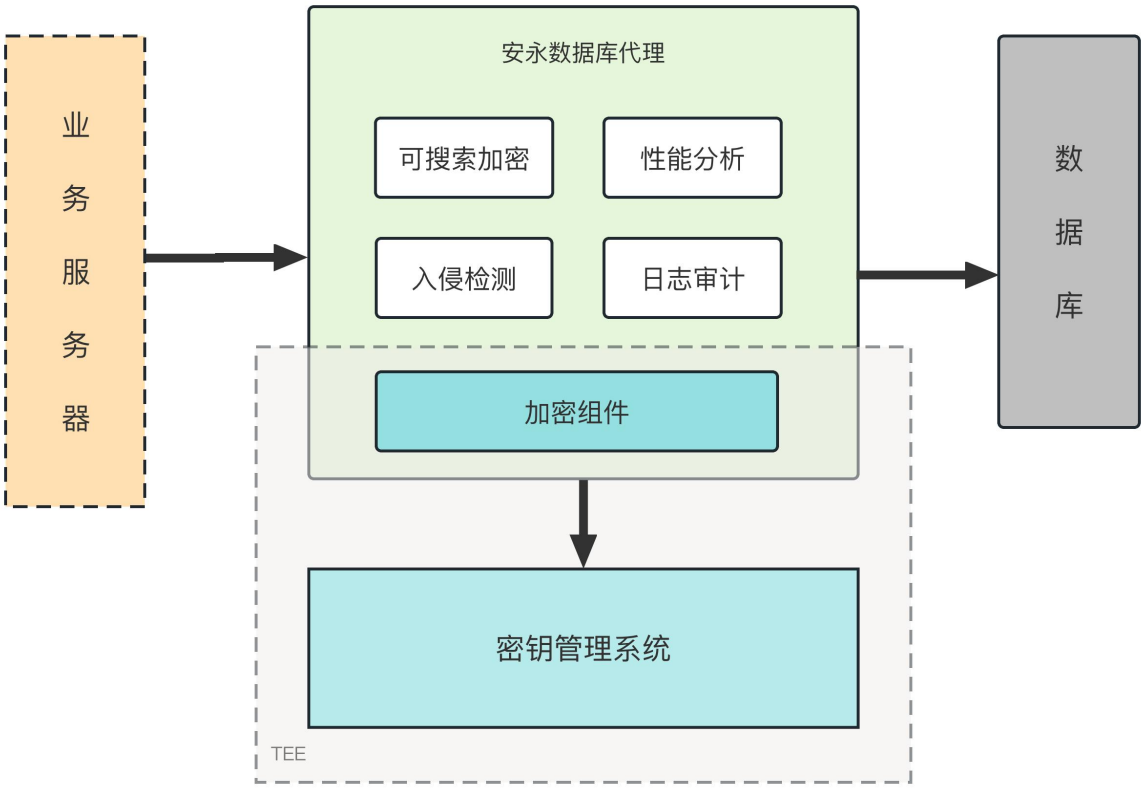
合肥安永信息科技有限公司

目录

安永数据库加密系统说明书

一、产品介绍

安永数据库加密系统是基于可搜索加密(Searchable Encryption)、硬件可信执行环境（Trusted Execution Environment）、透明加密技术等实现敏感数据加密存储的数据库防泄漏产品。系统支持使用加密算法 SM4 对敏感数据加密，支持列/表/库等不同细粒度的加密配置，可应用于关系型数据库的结构化数据加密，有效满足各类用户的等保、分保测评、数据安全防护需求。



1．产品优势

（1）基于 TEE 技术

密钥管理及使用限定 TEE 内部，密钥不明文存在于内存中，确保密钥安全。

（2）芯片级密钥安全

基于芯片级的密钥保护技术，保证密钥安全。

（3）高效加密索引

独有且经过工程化验证的可搜索加密技术，确保加密后的高效索引，解决行业痛点。

（4）独立密钥管理

支持软件、密码机、加密卡等多种密钥生成方式，加密密钥独立生成、独立管理。

2．功能特点

（1）动态加解密

加解密以智能化对存储在数据库中的数据进行自动实时灵活的动态加密解密，不需要人为干涉，完全实现对数据的安全存储、安全使用，防止信息涉密。

（2）高容灾能力

支持加密系统双机部署，主从互备，防止主机故障、网络故障、程序故障引起的业务损失。通过分批次加解密、数据校验保护、数据库运行状态监控、预加密验证、备份恢复等一系列可靠性保护技术，确保数据加解密过程中的安全、可靠和可恢复性。

（3）全透明访问

加密后对外部保持全透明特性，支持查询、插入、更新、删除等操作，应用系统无需改造。客户语句操作和应用程序不需修改，对应用开发接口全面透明，常用第三方管理工具可正常使用，对存储过程和函数透明。

（4）独立权控

提供针对加密数据的权限控制，从客户端的程序、IP、时间、星期多个维度限制部分用户的增删改查操作，防止敏感数据泄露。对授权用户提供无感知的自动加解密和透明访问，对非授权用户提供独立于数据库权限体系之外的第三方独立访问控制。

（5）支持多种数据类型和平台

可支持 ***Oracle***、 ***MySQL***、***PostgrelSQL*** 等多种数据库类型，支持 Linux 操作系统。

3．应用价值

（1）防黑客拖库

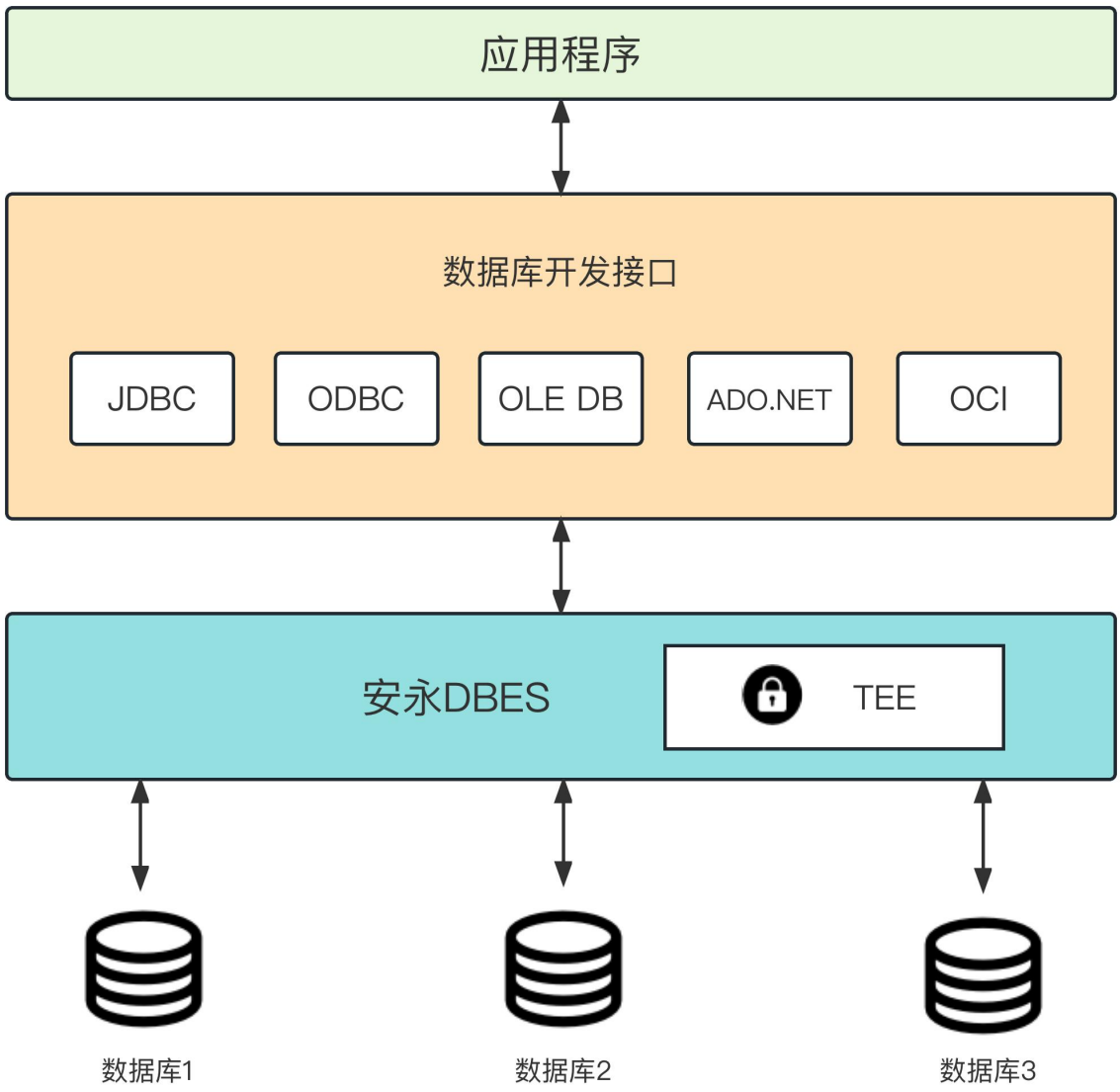
通过数据库透明加密产品对敏感数据进行加密存储，确保他人即使拿到了数据文件，也无法获取敏感信息。可有效防止黑客利用数据库漏洞攻击入侵数据库，最终针对数据库中的数据进行拖库。

（2）防越权访问

数据库维护人员，一般拥有 DBA 权限，可获取数据库中所有的敏感信息，存在内部人泄露风险，不符合安全规范。通过独立的、增强的权限控制、三权分立机制，确保任何用户在没有获得密文权限时无法获取数据库敏感信息。

4. 部署模式

数据加密产品采用旁路部署模式，与客户数据库系统路由可达即可，现有系统及网络拓扑均无需改动。外置设备部署完毕之后，系统功能开启后会自动在客户数据库系统中部署，无需人工干预，简便快速。



二、SGX 映像版本介绍

我们希望通过这个平台，与用户们共同学习、交流，共同探索可信执行环境的未来发展。我们欢迎用户们随时提出宝贵的建议和反馈，您的参与对我们至关重要。我们将始终倾听每一位用户的声音，持续优化和改进 SGX 版本，以更好地服务于用户的需求。

1．加密模块介绍

（1）用法以及参数详解

项目分为两个模块，密钥管理程序以及数据库加密代理程序。

- ***crypto_gm*** 是安永数据库加密系统密钥管理程序，负责系统密钥的初始化以及备份。

路径：*./bin/crypto_gm*

可用选项包括：

参数	功能
-i	生成密钥
-h, --help	参数目录
--sym	以对称方式（导入/导出）密钥
--asym	以非对称方式（导入/导出）密钥
-d, --import	导入密钥
-e, --export	导出密钥
-p, --pin	用户配置的 PIN 值
--client_id	配置客户端 id
--file	密钥文件
--password	AES128 密钥，HexString 形式
--pubkey	SECP256R1 公钥，HexString 形式

下面是生成密钥，导入、导出对称密钥，以及导入、导出非对称密钥的示范代码：

初始化系统密钥:

```
// -i 初始化密钥  --pin 用户配置的 PIN 值  --client_id 配置客户端 id
./crypto_gm -i --pin 12345678 --client_id testid
```

以对称方式导出密钥：

```
// -e 导出密钥  --sym 对称加密，采用 AES128 算法  --pin 用户配置的 PIN 值  --client_id 配置客户端 id  --password AES128 密钥，HexString 形式 --file 密钥文件
./crypto_gm -e --sym --pin=12345678 --client_id=testid \
--password=f79727ef29d0fe792c196c4bafcb3fb9 --file key.sym
```

以对称方式导入密钥:

```
// -d,--import 导入密钥  --sym 对称加密，采用 AES128 算法 --pin 用户配置的 PIN 值 --client_id 配置客户端 id  --password AES128 密钥，HexString 形式 --file 密钥文件
./crypto_gm -d --sym --pin=12345678 --client_id=testid \
--password=f79727ef29d0fe792c196c4bafcb3fb9 --file key.sym
```

以非对称方式导出密钥:

```
// -e 导出密钥  --asym 非对称加密，采用 SM2 算法 --pin 用户配置的 PIN 值 --client_id 配置客户端 id  --pubkey SECP256R1 公钥，HexString 形式  --file 密钥文件
./crypto_gm -e --asym --pin=12345678 --client_id=testid1 \
--pubkey=42A48FEB55BBFF2F8E310B2D77D98D7125B350A97A4AFC95EBCC4E64F7B\
99D5A030C7E966819B99D72F9DA1FBAA9C27B2E89ACD846B5FACC33CC3ACDADDA4DD7 \
--file keya.sym
```

以非对称方式导入密钥:

```
//  -d,--import 导入密钥  --asym 非对称加密，采用 SM2 算法 --pin 用户配置的 PIN 值 --client_id 配置客户端 id  --file 密钥文件
./crypto_gm -e --sym --pin=12345678 --client_id=testid \
--password=f79727ef29d0fe792c196c4bafcb3fb9 --file key.sym
```

• ***db_shield-server*** 安永数据库加密系统的主执行文件。为了确保系统能够根据您的需求成功启动，我们提供了以下参数供您选择和配置。

路径：*./bin/db_shield-server*

可用选项包括:

参数	功能
--postgresql_enable	启用此参数以激活代理功能，从而对 PostgreSQL 数据库进行加密和解密操作。
--mysql_enable	启用此参数以激活代理功能，从而对 MySQL 数据库进行加密和解密操作。
--tee_enable	SGX 可信执行环境的必选参数。选择此参数表示您希望在 SGX 可信执行环境中启动程序。
--db_port	指定数据库的通讯端口号。
--db_host	指定您的数据库的 IP 地址。
--encryptor_config_file	提供配置文件的路径，该文件描述了加解密操作的具体内容和目标数据库表。
--client_id	为客户端分配的唯一标识号。
--incoming_connection_host	设置加密代理的启动 IP 地址。默认情况下，这个地址是 0.0.0.0。
--incoming_connection_port	指定加密代理服务器的通讯端口号。默认情况下，这个端口是 9393。

使用 ***crypto_gm*** 初始化密钥后，可以通过以下代码启动安永数据库加密系统：

```
./db_shield-server --mysql_enable --tee_enable --db_port=3306 \
--db_host=127.0.0.1 --encryptor_config_file=search_test.yml \
--client_id="testid"
```

三、快速开始

1. 快速启动安装脚本

1. 运行 ./install.sh
2. 阅读 ReadMe.txt
3. (可选) 根据需求调整./crypro_gm.sh 脚本内参数
4. (可选) 根据需求调整./server.sh 脚本内参数
5. (可选) 根据需求编写数据库加密配置文件,例如 ./bin/test_search.yml
6. 运行./crypro_gm.sh 生成密钥
7. 运行 ./server.sh 以默认配置启动服务

2. 测试程序

（1）预备工作

本文档采用 PHP 文件对 MySQL 数据库的操作作为测试实例。在测试前，请确保已安装 PHP 及其相关依赖：

```
//对于 MySQL 数据库：
sudo apt install php php-mysql

//对于 PostgreSQL 数据库：
sudo apt install php php-pgsql
```

（2）设计并且创建加密表

对需要加密的表进行设计，需要加密的列必须要定义为 blob 类型。

如表 test 中，name_encrypted 与 age_encrypted 列需要加密，因此表 test 在创建时，需要将这两列定义为 blob 类型。

还需配置 ayx_name_encrypted_searchindex 字段代表 name_encrypted 字段可进行可搜索加密，类型也为 blob。

列名	是否加密	类型
name_encrypted	是	blob
name_unencrypted	是	blob
ayx_name_encrypted_searchindex	是	blob

```
CREATE SCHEMA IF NOT EXISTS test;
CREATE TABLE IF NOT EXISTS test.test (
`id` int(11) NOT NULL AUTO_INCREMENT,
`name_encrypted` blob DEFAULT NULL,
`name_unencrypted` blob DEFAULT NULL,
`age_encrypted` blob DEFAULT NULL,
`age_unencrypted` blob DEFAULT NULL,
`ayx_name_encrypted_searchindex` blob,
PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

（3）加密代理服务配置(提供默认配置文件)

配置在启动过程中的 encryptor_config_file 参数，指定为 test_search.yml 文件。文件内容如下:

```
#默认参数
defaults:
  #指定加解密方式
  crypto_envelope: dbshieldblock
  #默认可搜索加密子串长度，5 代表可搜索加密最小的子串长度为 5
  substring_length: 5
#需要加密的数据库
schemas:
  #需要加密的表
  - table: test
    #表的列名
    columns:
      - id
      - name_encrypted
      - name_unencrypted
      - age_encrypted
      - age_unencrypted
    #需要加密的列名
    encrypted:
      #列名
      - column: name_encrypted
        #返回结果的类型 str 代表 string int32 代表 int 的 32 位类型
        data_type: "str"
        #是否可以进行可搜索加密操作，true 代表是，不配置默认为 false，为 true 对应的
ayx_列名_searchindex 字段会有值
        searchable: true
        #可单独配置该列的最小可搜索子串长度，未设置默认为默认参数里所配置的值
        substring_length: 2
      - column: age_encrypted
        data_type: "int32"
```

（4）PHP 文件编写（提供默认测试文件）

```
<?php
function test($server, $port)
{
   echo "$server:$port\n";
   $conn = @mysqli_connect($server, 'root', '123456', 'test', $port);
   while (!$conn){
     sleep(1);
     $conn = @mysqli_connect($server,'root','123456','test',$port);
  }
  $conn->query('CREATE SCHEMA IF NOT EXISTS test;');
  $conn->query("CREATE TABLE IF NOT EXISTS test.test (
                  id int(11) NOT NULL AUTO_INCREMENT,
                  name_encrypted blob DEFAULT NULL,
                  name_unencrypted blob DEFAULT NULL,
                  age_encrypted blob DEFAULT NULL,
                  age_unencrypted blob DEFAULT NULL,
                  ayx_name_encrypted_searchindex blob,
                  PRIMARY KEY(id))ENGINE=InnoDB DEFAULT
                  CHARSET=utf8mb4;");
   if($port==3306){
   echo "直接使用数据库查询结果为:\n";
  $sql="select convert(name_encrypted using utf8mb4),convert(name_unencrypted
using utf8mb4),convert(age_encrypted using utf8mb4),convert(age_unencrypted
using utf8mb4) from test.test limit 500;";
   }else{
   echo "连接加密代理，并向表中插入数据:('test','test','12','12')\n";
   $conn->query("INSERT INTO
test.test(name_encrypted,name_unencrypted,age_encrypted,age_unencrypted)
VALUES('test','test','12','12')");
   echo "使用加密代理查询结果为:\n";
   $sql="select * from test.test limit 500 ";
 }
 $res=$conn->query($sql);
 if($res) {
  $rows=$res->fetch_all();
  print_r($rows);
 }else{
  echo mysqli_error($conn).':'.mysqli_error($conn);
 }
 mysqli_free_result($res);
 $conn->close();
  echo PHP_EOL;
}
test('127.0.0.1','9393');
test('127.0.0.1','3306');
```

（5）执行 PHP 文件

```
php test.php
```

（6）观察结果

通过 Go 代理可以直观看到原文，而通过 MySQL 数据库只能查看未加密字段。

（7）数据库查询验证

```
mysql -uroot -p
use test;
select * from test \G;
```

执行模糊搜索

```
select * from test where name_encrypted like 'te%';
```

从查询结果可见，相同数据加密后发生变化，但解密后数据保持一致，实现了透明加密。而 Go 代理能够实现模糊搜索，而原生数据库则不能。

① 常见问题(FQA)

2．-bash: Permission denied（权限问题）

```
// 给 start_server.sh 以及 crypto_gm 增加可执行权限
sudo chmod u+x start_server.sh
sudo chmod u+x start_crypto_gm.sh
sudo chmod u+rwx ./bin/enclave/enclave.signed
sudo chmod u+x ./bin/getlicense
sudo chmod u+x ./bin/db_shield-server
sudo chmod u+x ./bin/crypto_gm
```

3．FATA[0000] Please Make Sure Start In Azure with subscribed image !<nil>
请在 Azure 市场订阅的虚拟机映像产品中，启动该服务。

4．Failed to open Intel SGX device.
请使用推荐配置，或确保您的 *sgx* 驱动已启用。
运行 ***ls /dev/sgx**** 确保能够检索到驱动

五、联系我们

公司：合肥安永信息科技有限公司

地址：安徽省合肥市高新区创新大道 2800 号 J1 栋 A 座 1101 室

网站：www.anyong.net

邮箱：support@anyong.net

电话：+86 18055100335