opengauss-全密态数据库

一,简介

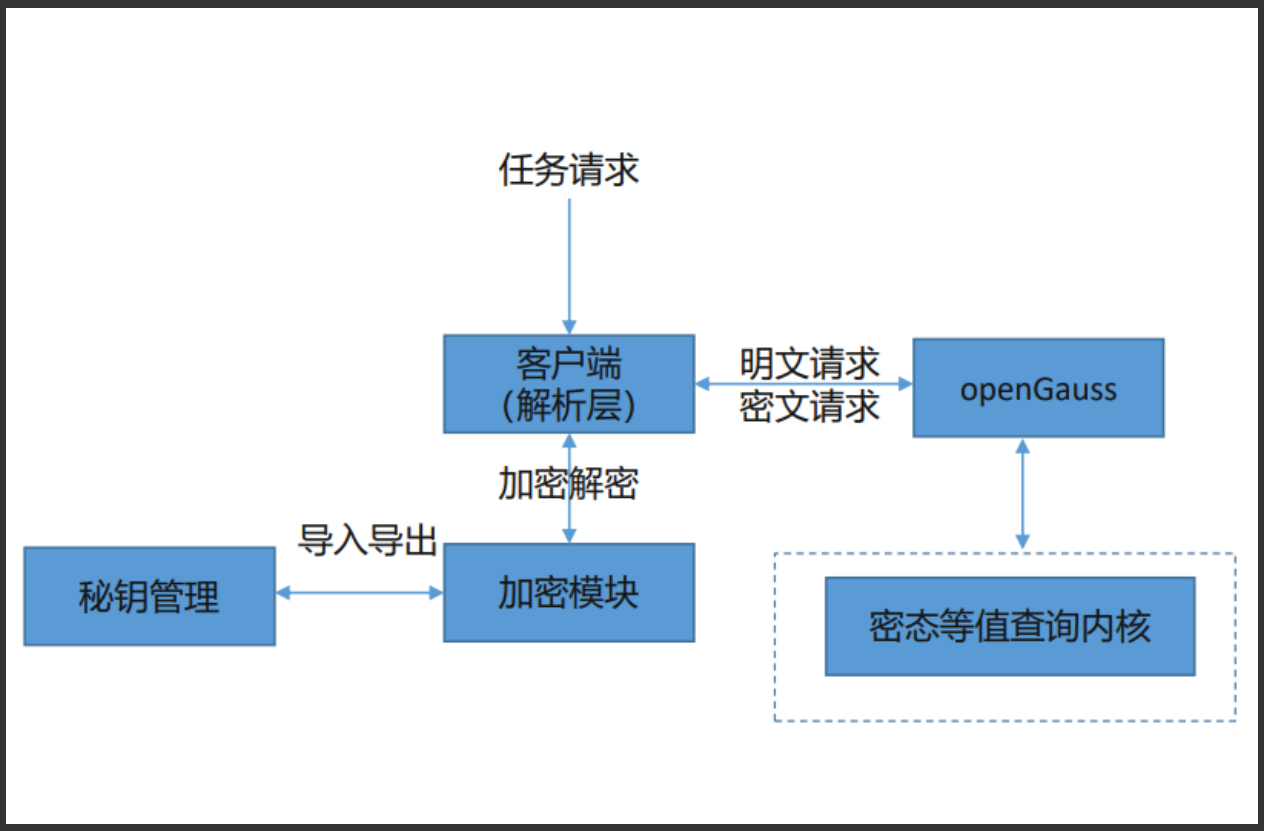
1.密态数据库意在解决数据全生命周期的隐私保护问题，使得系统无论在何种业务场景和环境下， 数据在传输、运算以及存储的各个环节始终都处于密文状态。

2..当数据拥有者在客户端完成数据加密并发送给服务端后，在攻击者借助系统脆弱点窃取用户数据 的状态下仍然无法获得有效的价值信息而起到保护数据隐私的能力。

3..由于整个业务数据流在数据处理过程中都是以密文形态存在，通过全密态数据库，可以实现：

- 保护数据在云上全生命周期的隐私安全，无论数据处于何种状态，攻击者都无法从数据库服务端获取有效信息。
- 帮助云服务提供商获取第三方信任，无论是企业服务场景下的业务管理员、运维管理员，还是消费者云 业务下的应用开发者，用户通过将密钥掌握在自己手上，使得高权限用户无法获取数据有效信息。
- 让云数据库借助全密态能力更好的遵守个人隐私保护方面的法律法规。

二,全密态等值查询流程



1. 三层秘钥管理机制：根秘钥，主秘钥和列加密秘钥。

2. 客户端完成数据的加密和解密，服务器完成密态数据计算。
3. 不需要加密的字段仍然是明文处 理。

三,动态数据脱敏机制

1. 数据隐私保护是数据库安全所需要具备的安全能力之一，可以在一定程度上限制非授权 用户对隐私数据的访问，保证隐私数据安全。

2.动态数据脱敏机制是一种通过定制化制定脱敏策略从而实现对隐私数据保护的一种技术，可以有效地在保留原始数据的前提下解决非授权用户对敏感信息的访问问题。

3.当管理员指定待脱敏对象和定制数据脱敏策略后，用户所查询的数据库资源如果关联到对应的脱敏策略时，则会根据用户身份和脱敏策略进行数据脱敏，从而限制非授权用户对隐私数据的访问。

四,脱敏策略内容

1.脱敏方式(Masking Function)，是指该脱敏策略使用何种方式对目标字段进行脱敏，目前openGauss预置了7种脱敏方式：creditcardmasking、 basicemailmasking、fullemailmasking、alldigitsmasking、shufflemasking、randommasking、maskall，分别适用于不同的脱敏场景

2.脱敏对象(Resource Label)是指脱敏策略生效时作用的对象集合（LABEL），若查询目标字段存在于LABEL中，则该字段将会根据脱敏策略进行敏感数据脱敏，openGauss动态数据脱敏特性支持对仅包含数据列的LABEL进行脱敏。
过滤器，指出脱敏策略在何种用户场景下生效，主要涉及USER（用户名）、APP（用户登录客户端名称）、IP（用户所处的ip）。当查询用户同时满足Masking Filter所指定的阈值时，数据脱敏策略才会生效。

五,触发脱敏策略

当系统接收到查询命令时，security_plugin将在解析器中拦截语义分析生成的查询树（Query），首先根据用户登录信息（用户名、客户端、IP）筛选出满足用户场景的脱敏策略。由于脱敏策略是基于（仅包含表列的）资源标签配置的，因此需要判断查询树的目标节点是否属于某个资源标签，然后将识别到的资源标签与脱敏策略相匹配，根据策略内容将查询树目标节点改写，最终将查询树返还给解析器。
security_plugin模块由于内置查询树脱敏方式，数据访问者不会感知内置安全策略重写查询树的过程，如同执行普通查询一样去访问数据，同时保护数据隐私。

脱敏函数名	示例
creditcardmasking	'4880-9898-4545-2525' 将会被脱敏为 'xxxx-xxxx-xxxx-2525'，该函数仅对后4位之前的数字进行脱敏
basicemailmasking	'abcd@gmail.com' 将会被脱敏为'xxxx@gmail.com'，对出现第一个'@'之前的文本进行脱敏
fullemailmasking	'abcd@gmail.com' 将会被脱敏为 'xxxx@xxxxx.com'，对出现最后一个'.'之前的文本（除'@'符外）进行脱敏
alldigitsmasking	'alex123alex' 将会被脱敏为 'alex000alex'，仅对文本中的数字进行脱敏
shufflemasking	'hello word' 将会被随机打乱顺序脱敏为 'hlwoeor dl'，该函数通过字符乱序排列的方式实现，属于弱脱敏函数，语义较强的字符串不建议使用该函数脱敏
randommasking	'hello word' 将会被脱敏为 'ad5f5ghdf5'，将文本按字符随机脱敏
maskall	'4880-9898-4545-2525' 将会被脱敏为 'xxxxxxxxxxxxxxxxxx'

六,实操示例

1. 打开内置安全策略[默认off]

```
[omm@lab01 ~]$ gs_guc reload -N all -I all -c "enable_security_policy=on"
[omm@lab01 ~]$ gsql -d postgres -p 26000 -c "show enable_security_policy ;"
 enable_security_policy------------------------ on
```

2. 创建测试表及数据

- 创建测试表person

```
create table person(id int primary key,name varchar(20),creditcard varchar(20),address varchar(50));
insert into person values(1,'张三','1234-4567-7890-0123','huoyue Mansion, No. 98, 1st Fuhua Street');
insert into person values(2,'李四','1111-2222-3333-4444','Futian District, Shenzhen City');
select * from person;+----+------+---------------------+-------------------------------------------+
| id | name |     creditcard      |                address                    |
+----+------+---------------------+-------------------------------------------+
|  1 | 张三 | 1234-4567-7890-0123 | huoyue Mansion, No. 98, 1st Fuhua Street |
|  2 | 李四 | 1111-2222-3333-4444 | Futian District, Shenzhen City           |
+----+------+---------------------+-------------------------------------------+
```

- 创建测试表orders

```
create table orders(id int primary key,pid int,customername varchar(20),order_no int,email varchar(50));
insert into orders values(1,1,'李雷',13002345,'654321@qq.com');
insert into orders values(2,1,'韩梅',13001234,'testdb@huawei.com');
insert into orders values(3,2,'Jerry',13009876,'test123@google.com');
select * from orders;
+----+-----+--------------+----------+
| id | pid | customername | order_no |
+----+-----+--------------+----------+
|  1 |   1 | 李雷         | 13002345 |
|  2 |   1 | 韩梅         | 13001234 |
|  3 |   2 | Jerry        | 13009876 |
+----+-----+--------------+----------+
```

3. 策略配置

- **创建资源标签**【对表的敏感字段添加资源标签(需要拥有poladmin权限)】

```
create resource label creditcard_label add column(person.creditcard);
create resource label customer_label   add column(orders.customername);
create resource label email_label      add column(orders.email);
create resource label id_label         add column(orders.id);
create resource label order_no_label   add column(orders.order_no);
create resource label pid_label        add column(orders.pid);
```

- **创建脱敏策略**

```
-- 语法:
CREATE MASKING POLICY policy_name masking_clause [, ... ] [ policy_filter_clause ] [ ENABLE | DISABLE ];
where masking_clause can be:
masking_function ON LABEL(label_name [, ... ],*)
where masking_function can be:
{ maskall | randommasking | creditcardmasking | basicemailmasking | fullemailmasking | shufflemasking | alldigitsmasking }
```

```
where policy_filter_clause can be:
FILTER ON { ( FILTER_TYPE ( filter_value [, ... ],* ) ) [, ... ],* }
where FILTER_TYPE can be:
{ APP | ROLES | IP }

-- 创建策略一【脱敏方式：maskall】
策略名：mask_card_pol
针对用户：user1
针对IP：192.168.0.99
针对应用：gsql
脱敏方式：creditcardmasking
create masking policy mask_card_pol
       creditcardmasking on label (creditcard_label)
       filter on roles('user1') ,IP('192.168.0.99'),APP('gsql');
-- 小缺陷：测试发现应用程序无法识别"Data Studio"，这个APP列表待完善,或者使用方法待说明

-- 创建策略二：【脱敏方式：maskall】
create masking policy mask_name_pol maskall on label(customer_label);

-- 创建策略三：【脱敏方式：randommasking】
create masking policy mask_id_pol randommasking on label(id_label);

-- 创建策略四：【脱敏方式：basicemailmasking】
create masking policy mask_email_pol basicemailmasking on label(email_label);

-- 创建策略五：【脱敏方式：alldigitsmasking 】
create masking policy mask_order_no_pol alldigitsmasking on label(order_no_label);

-- 创建策略六：【脱敏方式：shufflemasking 】
create masking policy mask_pid_pol shufflemasking on label(pid_label);
```

- **脱敏效果测试**

```
[omm@lab01 ~]$ gsql -d mydb -p 26000 -h 192.168.0.99 -U user1 -r
mydb=> select * from person;
 id | name |     creditcard      |                address
----+------+---------------------+---------------------------------------
  1 | 张三 | xxxx-xxxx-xxxx-0123 | huoyue Mansion, No. 98, 1st Fuhua Street
```

```
 2 | 李四 | xxxx-xxxx-xxxx-4444 | Futian District, Shenzhen City
mydb=# select * from orders;
 id | pid | customername | order_no |       email
----+-----+--------------+----------+--------------------
  0 |   0 | xx           |        0 | xxxxxx@qq.com
  0 |   0 | xx           |        0 | xxxxxx@huawei.com
  0 |   0 | xxxxx        |        0 | xxxxxxx@google.com
```