**金证股份•<**账户系统-国密改造**>**

**KBSS\_统一账户系统国密改造方案**

|  |  |  |  |
| --- | --- | --- | --- |
| **公司名称** | 深圳市金证科技股份有限公司 | **文档编号** |  |
| **文档名称** | KBSS\_统一账户系统国密改造方案 | **文档版本** | 1.0 |
| **起 草** | 李永洪 | **起草日期** | 2018-04-22 |
| **审 批** |  | **审批日期** |  |

**修订历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 日期 | 状态 | 修订人 | 摘要 |
| 1.0 | 2018-04-22 | C | 李永洪 | 初稿 |
|  |  |  |  |  |
|  |  |  |  |  |

状态标识：C –Created A- Added M - Modified D - Deleted

目录

[1 引言 6](#_Toc514656061)

[1.1 目的和范围 6](#_Toc514656062)

[1.2 背景 6](#_Toc514656063)

[2 需求概述 6](#_Toc514656064)

[3 设计概述 6](#_Toc514656065)

[3.1 设计策略 6](#_Toc514656066)

[3.1 系统现状（改造前） 7](#_Toc514656067)

[3.2 时序图(改造前) 8](#_Toc514656068)

[3.3 操作员密码 8](#_Toc514656069)

[3.4 交易密码 9](#_Toc514656070)

[3.5 资金密码 10](#_Toc514656071)

[3.6 银行密码 10](#_Toc514656072)

[4 改造方案 11](#_Toc514656073)

[4.1 系统现状(改造后) 12](#_Toc514656074)

[4.2 时序图（改造后） 13](#_Toc514656075)

[4.3 主要业务流程说明 14](#_Toc514656076)

[4.3.1 用户登陆 14](#_Toc514656077)

[4.3.2 修改密码 15](#_Toc514656078)

[4.3.3 重置密码 16](#_Toc514656079)

[4.4 关键变更支持 16](#_Toc514656080)

[4.4.1 外围算法库encrypt.dll 16](#_Toc514656081)

[4.4.2 柜台算法库KDStorageCrypt.dll 16](#_Toc514656082)

[4.4.3 校验逻辑修改 17](#_Toc514656083)

[4.4.4 柜台算法库CITICS\_CE.dll 17](#_Toc514656084)

[4.4.5 柜台算法库KDEncodeCli.dll 17](#_Toc514656085)

[4.4.6 金仕达贵金属系统算法库KSTEncryptd.dll 17](#_Toc514656086)

[4.4.7 顶点OTC系统对接 17](#_Toc514656087)

[4.5 新增功能 17](#_Toc514656088)

[5 数据表设计 17](#_Toc514656089)

[6 LBM接口清单 18](#_Toc514656090)

[7 参考文件 18](#_Toc514656091)

# 引言

## 目的和范围

本文档用于描述统一账户系统如何设计实现：账户系统底层加密算法库，迁移至金证国密加密算法的改造方案。

## 背景

为了保障商用密码安全，国家商用密码管理办公室制定了一系列密码标准。根据证信办[2014]157号文件《证券期货业国产密码应用推进规划方案》和国务院办公厅发布《金融领域密码应用指导意见》要求，为配合国家对金融安全领域的升级，我司提出了对自身系统的改造计划，推出了金证国密算法库，统一账户系统通过切换新的算法库，将现有的加密体系切换到新的国密加密体系，完成对加密算法的升级。

# 需求概述

根据公司要求，在2018年6月前完成金证系统国密改造，2018年12前实现国产密码在网上证券交易系统全面应用。为此，统一账户系统目标在6月前完成国密改造，与各系统联调完成，并发布生产版本，版本同时支持新老两种密码加密方式。

# 设计概述

## 设计策略

本次改造总的设计原则是与统一账户系统对接的周边系统，仅通过增加底层加密算法库KDStorageCrypt.dll、更新encrypt.dll外围通讯加密算法库完成升级，业务接口不做任何改动。统一账户系统本身要实现升级前后的兼容处理，一方面在检查外部系统传入的密码时，要能正确加解密，并适配后台存量密码的加密算法进行正确转换，另一方面，对存量的客户数据，要能做到分阶段、分批次，增量切换到国密算法。具体策略如下

* 支持向下兼容、逐步切换

新老算法由于存在过渡期，所以支持向下兼容逐步切换

* 支持批量切换

支持柜员通过批量菜单全部更新客户存储密码

* 支持存储加密个性化算法

支持不同的券商采用不同的加密算法，以及个性化密钥

* 统一认证模式，修改密码由统一认证校验密码

当前版本修改柜员密码时，同步统一认证；后续修改密码时远程调用统一认证的功能，由统一认证校验。

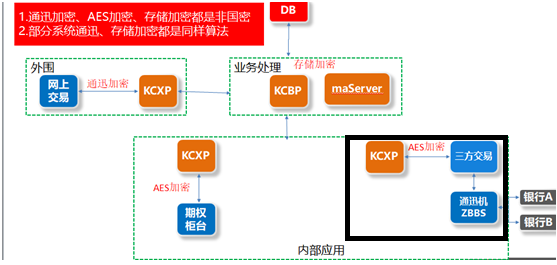
* 外围系统0改动

切换算法的过程中外围程序不需要进行程序的修改

## 系统现状（改造前）

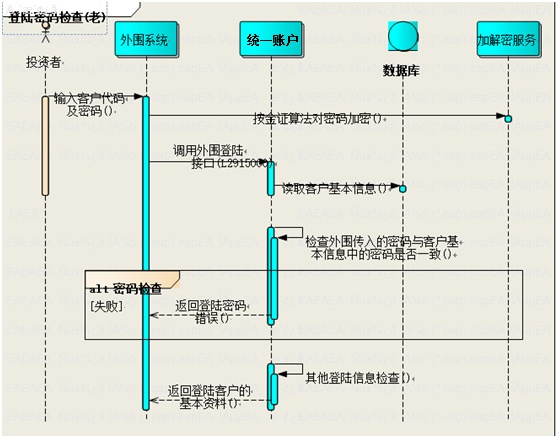
在统一账户系统(KBSS架构)中，大部分加解密算法都是对应国密算法的SM4对称加密算法。

|  |  |  |  |
| --- | --- | --- | --- |
| **DLL名称** | **DLL版本** | **导出函数** | 用途 |
| encrypt.dll |  | kbss\_comencrypt | 用于外围通信加密 |
| kbss\_comdecrypt | 用于外围通信解密 |
| AES\_Encrypt1 | 用于柜台系统、三方交易通信加密 |
| AES\_Decrypt1 | 用于柜台系统、三方交易通信解密 |
| MD5\_Digist | 未用到 |
| Base64\_Encode | 没用到 |
| Base64\_Decode | 没用到 |
| RC5\_Encrypt | 用于登陆生成票据 |
| RC5\_Decrypt | 用于登陆票据解密 |
| RSA\_Encrypt | 用于登陆生成票据 |
| RSA\_Decrypt | 没用到 |
| kbss\_encrypt1 | 用于数据存储加密 |
| kbss\_recrypt1 | 没用到 |
| kbss\_recrypt2 | 没用到 |



注:账户计划后续直接对接银行

## 时序图(改造前)



## 操作员密码

系统的操作员密码是存在数据库中，在统一账户系统本地校验用户登录；

具体内容如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **密码** | **系统** | **涉及业务** | **详细信息** | **备注** |
| 操作员  密码 | 统一账户系统 | 1、用户认证信息校验L0000506  用户身份注册L0000507  用户密码管理L1100009 | 1、操作员密码存在统一账户系统数据库中  2、操作员密码通过AES\_Decrypt1解密后再用kbss\_encrypt加密 | 本系统 |
|  |  |  |  |
| 统一认证 | 用户认证 | 启用统一认证时，到统一认证校验操作员密码 |  |
|  |  |  |  |

## 交易密码

交易密码在统一账户系统以存储加密方式将密文保存在数据库中；

具体如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **密码** | **关联系统** | **涉及业务** | **详细信息** | **备注** |
| 交易密码 | 统一账户系统 | 客户修改交易密码  客户重置交易密码 | 1、统一账户系统对资金密码进行解密再加密后传给业务系统交易 | 本系统 |
| 股票期权系统 | 客户修改交易密码  客户重置交易密码 | 资金密码加密后传入统一账户系统 | 关联系统 |
| 银行(暂未对银证) | 银行发起银行转证券  银行发起证券转银行  银行发起查询银行余额  银行发起银衍开户  银行发起银衍签约  银行发起银衍销户  银行发起同步客户资料  银行发起变更银行账号 | 资金密码由银行加密传入 | 发起方 |
| 三方交易网关(暂未对银证) | 银行发起银行转证券  银行发起证券转银行  银行发起查询银行余额  银行发起银衍开户  银行发起银衍签约  银行发起银衍销户  银行发起同步客户资料  银行发起变更银行账号 | 三方网关采用AES加密方式传送资金密码密文 | 网关 |
| 外围 | 外围登陆  外围密码修改  外围登陆（申万宏源） |  |  |
|  |  |  |  |  |

## 资金密码

资金密码在统一账户系统以存储加密方式将密文保存在数据库中；

具体如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **密码** | **关联系统** | **涉及业务** | **详细信息** | **备注** |
| 资金密码 | 统一账户系统 | 客户开立资金账户设置资金密码  客户修改资金密码  客户重置资金密码  券商发起银行转证券  券商发起证券转银行  券商发起查询银行余额  券商发起银衍开户  券商发起银衍签约、  券商发起银衍销户  银行发起查询银行余额  银行发起银衍开户  银行发起银衍签约  银行发起银衍销户 | 1、统一账户系统对资金密码进行解密再加密后传给业务系统交易 | 本系统 |
| 股票系统 | 客户修改资金密码  客户重置资金密码 |  | 关联系统 |
| 三方交易网关 | 银行发起银衍开户  银行发起银衍签约  银行发起银衍销户 | 三方网关采用AES加密方式传送资金密码密文 | 网关 |
|  |  |  |  |

## 银行密码

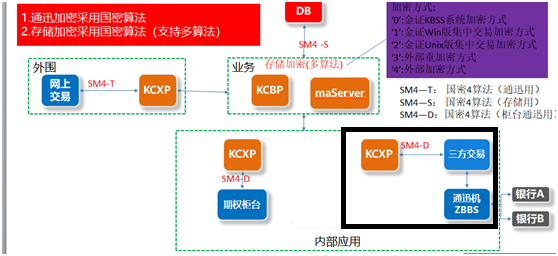
银行密码发起方有外围、账户系统，三方银衍网关，需要对银行密码进行加密操作，通过三方交易网关发到到银行，由银行进行校验。具体如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **密码** | **关联系统** | **涉及业务** | **详细信息** | **备注** |
| 银行密码 | 账户系统 | 券商发起银衍开户  券商发起银衍签约、  券商发起银衍销户  券商发起同步客户资料 | 账户系统进行银衍开销户时、同步客户资料、单账户结息归本同步银行时，使用AES\_Encrypt1加密银行密码 | 本系统 |
| 外围系统 |  | Web客户端等使用AES\_Encrypt1加密银行密码 | 前端 |
| 银行 | 券商发起查询银行余额  券商发起银衍开户  券商发起银衍签约、  券商发起银衍销户  券商发起同步客户资料 | 银行校验银行密码是否正常  银行发起业务将银行密码加密后，传入新三方交易网关 | 校验 |
| 三方交易网关 | 券商发起查询银行余额  券商发起银衍开户  券商发起银衍签约、  券商发起银衍销户  券商发起同步客户资料  银行发起查询银行余额  银行发起银衍开户  银行发起银衍签约  银行发起银衍销户  银行发起同步客户资料  银行发起变更银行账号 | 券商发起业务，网关对银行密码解密在加密后传给银行  银行发起业务，网关对银行密码进行解密加密后，发送给统一账户系统 | 关联系统 |

# 改造方案

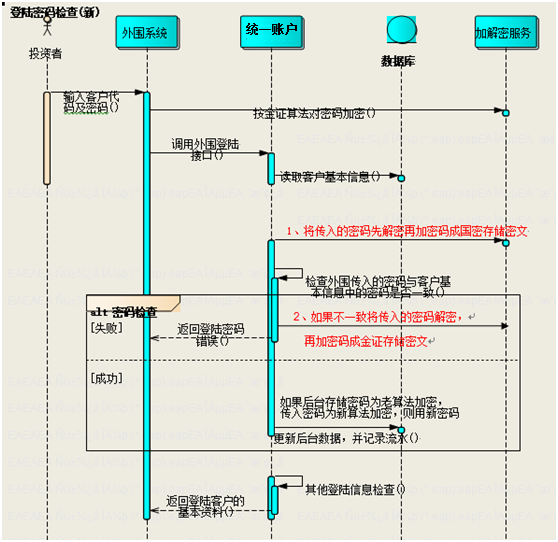
统一账户系统中采用了一个统一封装的加密算法库，目前算法库名称为encrypt.dll。建议由网关产品使用国密算法，封装一个国密的加密算法库，统一给产品部门使用。原来加解密函数名和出入参不变，业务系统调用原函数，变动无感知。在正式切换到国密算法后，后台数据库中存储的密码需要逐步切换到新的加密算法。

## 系统现状(改造后)



注:账户计划后续直接对接银行

## 时序图（改造后）



## 主要业务流程说明

### 用户登陆



### 修改密码



### 重置密码



## 关键变更支持

### 外围算法库encrypt.dll

将原有算法库encrypt.dll中的外围通信加解密（kbss\_comencrypt/kbss\_comdecrypt）封装为国密算法，出入参数不变。外围厂商只需更新encrypt.dll动态库，仍然调用原来的加密码函数。

### 柜台算法库KDStorageCrypt.dll

提供新算法库KDStorageCrypt.dll，封装外围通信加解密函数（kbss\_comencrypt/kbss\_comdecrypt）、柜台通信加解密函数（AES\_Decrypt1/AES\_Encrypt1）、存储加解密函数（kbss\_encrypt1/kbss\_decrypt1）为国密算法，出入参数不变。产品部门调用原来的加密码方法，只用使用KDStorageCrypt.dll密码库。另增加一组（kbss\_comencrypt0/kbss\_comdecrypt0）、柜台通信加解密（AES\_Decrypt0/AES\_Encrypt0）、存储加解密（kbss\_encrypt0/kbss\_decrypt0）能兼容旧版本。

### 校验逻辑修改

原逻辑：

1、先将通信密文解成明文

2、再把明文加密成金证的存储密文，同数据库客户信息中的密文比较，如果一致，校验通过。如果不一致，则报密码错误

新逻辑：

1、先调用密码识别函数

2、根据密码标识将通信密文解成明文

3、再把明文加密成国密存储密文后同数据库客户信息中的密码比较，如果一致，则返回成功。

4、如不一致，再将明文加密成金证旧的存储密文，同数据库客户信息中的密文比较，如果一致，则将数据库客户信息中的密文更新为国密，并写流水。如果不一致，则报密码错误

### 柜台算法库CITICS\_CE.dll

账户系统U版本中转BP“新一代UNIX版柜台加密函数CITICS\_CE.dll”/“KBSS版柜台解密函数encrypt.dll”替换为支持国密版本；

### 柜台算法库KDEncodeCli.dll

账户系统Win版BP“新一代Win版柜台加密函数KDEncodeCli.dll” /“KBSS版柜台解密函数encrypt.dll”替换为支持国密版本；

### 金仕达贵金属系统算法库KSTEncryptd.dll

KSTEncryptd.dll对接金仕达黄金交易系统交易密码/资金密码 加解密调整为KDStorageCrypt.dll国密通讯加密方式

### 顶点OTC系统对接

对接顶点OTC系统涉及交易密码和资金密码调整为KDStorageCrypt.dll国密通讯加密方式

## 新增功能

新增菜单，支持将后台数据库中存放的密码，批量转换为国密加密方式。对存量的客户数据，要能做到分阶段、分批次，增量切换到国密算法。

# 数据表设计

无

# LBM接口清单

**修改LBM清单：**

L1190038用户密码修改(个股期权)

L1190163一户通客户密码变更

L1190001用户密码维护

L0000505密码策略修改

L1100009用户密码修改

L1160482消费支付密码管理

L1160483消费支付密码挂失

L1160484消费支付密码验证

L1192069客户密码解锁

L1192100网厅用户密码管理

L1192163检查消费支付密码是否设置

L1100806一账通密码设置

99003101一账通密码设置

L0060510用户认证信息查询

L0000506用户认证信息校验

# 参考文件

《证券期货业国产密码应用推进规划方案》 证信办[2014]157号

《金融领域密码应用指导意见》

国密改造更新加密库:

加密库变更清单：

1、QueryType 功能描述：密文类型查询接口

int DLLEXPORT KBSS\_CALLTYPE QueryType(unsigned char \*pSrcData, int nSrcDataLen, char \*chType);

2、AES\_GM\_Encrypt1 功能描述：国密版AES加密算法,前后台加解密认证数据时用

void DLLEXPORT KBSS\_CALLTYPE AES\_GM\_Encrypt1(char \*p\_pszEncrResult, int p\_iSize, \_\_int64 p\_llKey, const char \* p\_pszEncrInfo);

3、AES\_GM\_Decrypt1 功能描述：国密版AES解密算法.前后台加解密认证数据时用

void DLLEXPORT KBSS\_CALLTYPE AES\_GM\_Decrypt1(char \*p\_pszDecrResult, int p\_iSize, \_\_int64 p\_llKey, const char \* p\_pszDecrInfo);

4、MD5\_GM\_Digist功能描述：国密版MD5报文摘要算法，此算法不可逆，用于通信过程中信息的防篡改

void DLLEXPORT KBSS\_CALLTYPE MD5\_GM\_Digist(char \* p\_pszDigResult, int p\_iSize, unsigned char p\_pszDigRetInt[16], unsigned char \* p\_pszDigInfo);

5、RC5\_GM\_Encrypt1，功能描述：国密版RC5加密算法，此算法用于前后台加解密保密键(验证码)。

void DLLEXPORT KBSS\_CALLTYPE RC5\_GM\_Encrypt1(char \* p\_pszEnResult, int p\_iSize, \_\_int64 p\_llKey, const char \* p\_pszEnInfo);

6、RC5\_GM\_Encrypt/RC5\_GM\_Decrypt国密版

// iEncryptType = 1 代表服务端加解密Key2。这时传入的pszKey与iKeySize没有作用，内部写死有密钥。

// iEncryptType = 2 其它情况加解密

void DLLEXPORT KBSS\_CALLTYPE RC5\_GM\_Encrypt(char \* p\_pszEnResult, int p\_iSize, const unsigned char \* p\_pszKey, int p\_iKeySize, int p\_iEncryptType, const char \* p\_pszEnInfo);

void DLLEXPORT KBSS\_CALLTYPE RC5\_GM\_Decrypt(char \* p\_pszDeResult, int p\_iSize, const unsigned char \* p\_pszKey, int p\_iKeySize, int p\_iDecryptType, const char \* p\_pszDeInfo);

7、kbss\_GM\_encrypt1，功能描述：数据存储加密算法：

int DLLEXPORT KBSS\_CALLTYPE kbss\_GM\_encrypt1(char \*p\_pszOutput, int p\_iFixedSize, const char \*p\_pszInput, const char \*p\_pszKey, int p\_nEncodeType);

8、kbss\_GM\_recrypt1 功能描述：数据存储重加密算法，此函数用于更换加密的密钥

int DLLEXPORT KBSS\_CALLTYPE kbss\_GM\_recrypt1(char \*p\_pszOutput, int p\_iFixedSize, const char \*p\_pszInput, const char \*p\_pszOldKey, const char \*p\_pszNewKey, int p\_nEncodeType);

9、kbss\_GM\_encrypt，功能描述：数据存储加密算法

void DLLEXPORT KBSS\_CALLTYPE kbss\_GM\_encrypt(char \*p\_pszOutput, int p\_iFixedSize, const char \*p\_pszInput, const char \*p\_pszKey);

10、kbss\_GM\_recrypt 功能描述：数据存储重加密算法，此函数用于更换加密的密钥

void DLLEXPORT KBSS\_CALLTYPE kbss\_GM\_recrypt(char \*p\_pszOutput, int p\_iFixedSize, const char \*p\_pszInput, const char \*p\_pszOldKey, const char \*p\_pszNewKey);

11、kbssCipherTextToGMCipherText 功能描述：密文转国密密文

void DLLEXPORT KBSS\_CALLTYPE kbssCipherTextToGMCipherText(char \*p\_pszOutput, int p\_iFixedSize, const char \*p\_pszInput, const char \*p\_pszKey, int p\_nEncodeType);

12、kbss\_GM\_comencrypt功能描述：国密通信加密算法，用于通信过程中关键信息的加密

void DLLEXPORT KBSS\_CALLTYPE kbss\_GM\_comencrypt(char \*p\_pszOutput, const char \*p\_pszInput, const char \*p\_pszKey);

13、kbss\_GM\_comdecrypt 功能描述：国密通信加密算法，用于通信过程中关键信息的解密

void DLLEXPORT KBSS\_CALLTYPE kbss\_GM\_comdecrypt(char \*p\_pszOutput, const char \*p\_pszInput, const char \*p\_pszKey);