The Information Security System of Commercial Bank Based on SOA

Ying'an CUI[1,2], Hui XIA[1]

[1]*School of Computer Science and Engineering, Xi'an University Of Technology, Xi'an, China*

[2]*School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China*

Email: *suchdaysuchpeople@126.com*

Abstract: This article presents a solution of the information security System of Commercial Bank which is based on SOA in data concentration environment and introduces the topology structure, main function, key technology of the information security system. It includes hierarchical group key management, data integrity service, datagram accuracy service and gives a concrete example of how to use DAC. The system has been successfully implemented and work stable .So it has significance reference value to similar system.

Keywords: information security; Service-Oriented Architecture; hierar-chical group key management; security engine

基于 SOA 的商业银行信息安全系统

崔颖安[1,2], 夏　辉[1]

[1]西安理工大学计算机科学与工程学院，西安，中国，710048

[2]西安交通大学电信学院，西安，中国，710048

Email: suchdaysuchpeople@126.com

【摘要】 本文提出了在全国数据大集中环境下基于 SOA 架构的商业银行信息安全解决方案。介绍了该系统的拓扑结构、主要功能、关键技术。包括分层密钥管理、数据完整性服务、数据真实性服务、报文准确性服务，并给出了 DAC 的应用实例。目前该系统已经投入使用，运行状况良好，对类似应用具有一定参考价值。

【关键词】 信息安全；面向服务体系结构；分层密钥管理；安全引擎

1 引言

我国银行业信息化始于２０世纪８０年代，经过30年的发展，已完成各类业务系统的建设及数据的全国集中处理，实现全业务的信息化运营与管理[1]。信息安全在银行信息化建设中占有重要地位，它是安全支付、规范管理的前提与保障。尤其是在全国数据集中的计算环境下，不但要解决数据存储安全、报文传输安全、客户身份鉴别等基础的信息安全问题，还要解决不同业务系统的个性化信息安全需求与集中式信息安全管理的矛盾[2]。SOA 可以将信息安全功能分解成细粒度的原子服务，通过对不同原子服务的组合构造出适合不同需求的个性化服务，这样既能解决信息安全系统集中管理的要求也能满足不同业务系统的个性化需求，是异构环境的最佳实践方法。本文以某全国性商业银行数据集中工程为背景，介绍了基于 SOA 信息安全系统的设计与实现。

2 银行信息安全系统设计

2.1 SOA 简介

面向服务的体系结构（Service-Oriented Architecture，SOA）是一种开放的、敏捷的、可扩展的、可联邦的、可组合的软件体系结构模型，它由高度自治的、高服务质量的、可互操作的、可发现的和可复用的服务构成，通过良好的接口和契约将服务组合起来实现特定的任务。接口采用中立的方式进行定义，它独立于实现服务的硬件平台、操作系统编程语言，能够以统一的方式在异构的环境中使用。SOA 由服务使用者、服务提供者、服务注册中心三部分构成[3-5]，结构如**图 1** 所示：

1）服务使用者可以是一个应用程序、一个软件模块或需要某一服务的另一服务,它发起对注册中心的查询，通过传输绑定协议，获得可以使用的服务；

978-1-935068-12-9

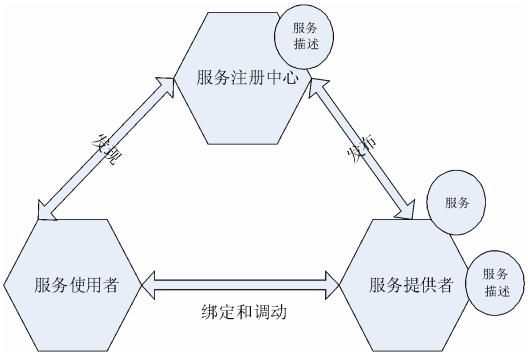

Figure 1. The architecture of SOA
图 1. SOA 的结构

2）服务提供者是一个可寻址的实体，它接受服务使用者的请求，将自身的接口契约发布到服务注册中心，以便服务使用者发现和访问。

3）服务注册中心是各类服务发现的管理者，由可用服务存储库和接口契约文档构成，经过验证的服务使用者通过查找服务接口获得可使用的服务。

2.2 系统设计

1) 拓扑结构

系统采用 SOA 中的 M-Lreopen 设计模式，服务使用者是各类业务系统，服务提供者是信息安全系统，业务系统与信息安全系统彼此独立，通过高速光纤网络相连。各类业务系统仅负责业务逻辑的处理，所有信息安全服务部署在信息安全系统中，包括分级密钥管理、数据完整性服务、实体鉴别服务等，业务系统以服务调用的方式访问信息安全系统。 调用过程如下：

a）业务系统将信息安全服务请求提交到安全引擎的 Rec Queue 中；

b）安全引擎首先进行服务合法性校验，若该服务请求与安全引擎具有合法的契约，则申请资源并脱离安全引擎上下文环境；

c）分析服务流程并创建服务线程，在服务线程中按照配置信息执行相应的服务；

d）服务执行完以后向安全引擎的 Sed Queue 队列中返回调用结果，安全引擎将响应信息返回调用者，系统拓扑结构如**图 2** 所示。

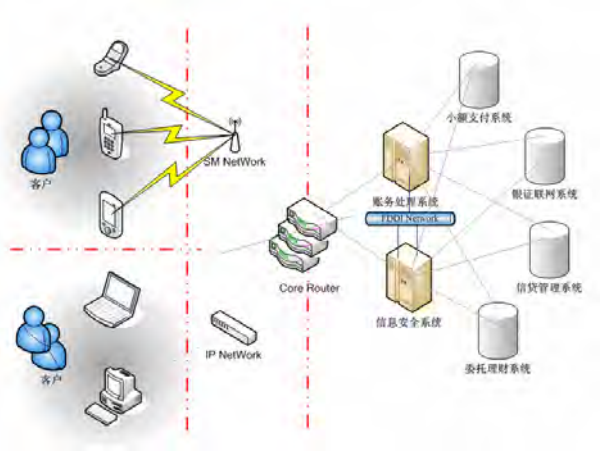

Figure 2. System topology
图 2.系统拓扑结构图

3 系统主要功能

1）分级密钥管理

按照统一管理，风险分散的原则，系统采用分级密钥管理方法。系统包括三类密钥：系统根密钥、发行密钥、传输密钥[6-7]。

系统根密钥在总行生成。由总行的四位高级管理者输入各自的指纹信息，通过对指纹信息的二值化处理与转换，产生一个初始密钥。而后用该密钥与 IC 卡进行随机杂凑生成新的密钥，该密钥作为系统根密钥。系统根密钥并不直接用于数据加密，它是用于生成各分支机构密钥的母密钥。

发行密钥包括省级分行和地市分行两级发行密钥。省级分行发行密钥先由省行四位高管输入各自的指纹信息，生成省级初始密钥，而后再使用该密钥与系统根密钥进行随机杂凑而成。地市分行的发行密钥与省行的发行密钥产生过程相似，不同之处在于它使用省行的发行密钥而不是系统根密钥进行杂凑，此外杂凑规则与省行发行密钥不同。

传输密钥用于报文完整性校验，该密钥存放在柜员 IC 卡或 Ukey 中，由省行统一制卡并完成该密钥的初始化。在柜员向业务系统注册时，使用 IC 卡或 Ukey 生成的随机数与系统时间戳、机构代码、操作员职工号等个性化因素散列生成新的传输密钥，而后以密文形式存放在网点前端共享内存中，同时更新 IC 卡或 Ukey 中的传输密钥并在原传输密钥的保护下同步后端数据库中存放的传输密钥。正常情况下，系统使用共享内存中的传输信息，若共享内存信息失效，则读取数据库中的信息并写入共享内存供后续处理使用。

层次化密钥管理符合大型信息安全系统的特点，其优点是通过不同层次的密钥衍生与绑定形成相互制约的安全格局，减少了密钥失窃的风险，提高了信息安

全系统的强度。

2）数据完整性服务

数据完整性是指系统确保信息准确和可靠的特性，即报文在存储过程中不被非法修改和破坏，保证数据一致性的能力[8-10]。在完成全国数据集中处理以后，各类账务信息集中存放在总行数据中心，因此数据完整性就成为信息安全要解决的首要任务。解决数据完整性的技术方法是 DAC（Data Authority Code），该方法的原理是在账务信息相关的数据表中增加 dac 字段，从表中提取需要保护的特定数据元构成一条初始信息，而后通过一系列的复杂变换与混淆形成一条信息摘要，用地市分行的发行密钥加密处理后再进行一系列复杂变换生成 DAC 存放到 dac 字段中。每当客户发生金融类交易时，重新计算该记录的 DAC，若结果不一致，则表示数据被非法修改，交易中断，这样就能发现账户信息的异动。

3）报文准确性服务

报文准确性服务是指对报文发送者和接收者之间报文始发源和内容有效性的验证，即报文在传输过程中不被非法修改和破坏，保证数据一致性的能力[8-10]。在完成全国数据集中处理以后，所有的金融类交易都需要对报文的准确性进行检查，验证报文准确性的技术方法是 MAC（Meessage Authority Code）。该方法的原理是在传输报文信息的尾部增加 mac 字段,从报文中提取需要保护的特定数据元构成一条初始信息，而后通过一系列复杂变换与混淆形成一条信息摘要，用柜员个人的传输密钥加密处理后再进行一系列复杂变换生成 MAC 附加在报文后。每当客户发生金融类交易时，报文发送者和接收者进行 MAC 双向校验，若结果不一致，表示数据在传输过程中被非法修改，不进行任何交易，系统提示报警。MAC 不负责报文信息的加密，报文信息加密由 VPN 完成，通过 VPN 完成报文的加、解密处理，可以充分利用硬件设备的特性，提高系统的整体性能。

4）实体鉴别服务

实体鉴别服务是指对服务调用者身份的鉴别，即确保服务调用者具有合法的操作权力和数据存取控制权限[8-10]。系统采用的实体鉴别服务主要包括两类方法：一类是密码校验，一类是物理校验。密码校验是取用户的若干个性化元素（例如个人客户的账号）与用户密码明文组合成初始信息，而后使用地市分行的发行密钥加密处理后再进行一系列复杂变换生成密码信息，该信息具有唯一性，确保一人一密，密文绝不重复（即使明文相同）。物理校验是另外一种实体鉴别方法，它是对密码校验的补充，使用 IC 卡和 Ukey 作为物理介质，用户只有通过物理介质和系统的双向认证，再配合密码校验才能通过鉴别服务。

4 关键技术

4.1 安全引擎

安全引擎的结构分为 4 层[12]，如**图 3** 所示。

配置管理层：包括安全引擎运行参数的设置、服务的定义与描述、服务组件的分发与部署、服务的重组与装配、运行状态的监控、服务调用者的配置等功能。

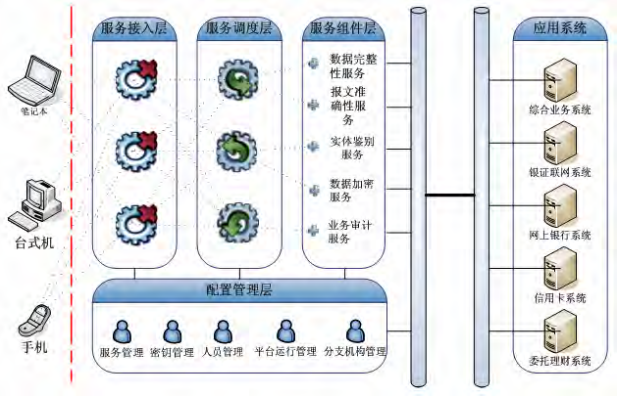

Figure 3. The architecture of security engine

图 3. 安全服务引擎的结构

服务接入层：负责信息安全服务请求的接入及服务合法性的校验，只有符合服务契约的请求才能提交后端激发相应服务。

服务调度层：在调用具体服务之前，由安全引擎分配资源，为其创建独立的线程资源及上下文交互环境，而后根据送达的业务标识码获得与之对应的服务信息（如服务的位置信息、外部接口、执行权限等），由服务引擎为其分配服务资源并激发服务。

服务组件层：服务组件层由物理服务和逻辑服务构成。物理服务是细粒度的原子服务，是最基础的信息安全服务单元；逻辑服务是用户根据自己的个性化需求使用原子服务组合而成的大粒度服务。服务使用定义良好的接口和契约进行描述，各类服务彼此独立，可以根据需求动态灵活调整。

分层设计一方面降低了软件开发的难度，应用系统的开发人员只需按照接口规约进行数据交换，不需

978-1-935068-12-9 © 2010 SciRes.

要了解任何信息安全的细节知识；另外，提高了系统的可维护性，只需通过简单的配置就可完成系统维护和扩充，不需要修改任何程序，将安全引擎与具体服务的耦合性降到了最低。

5 服务封装实例

本系统采用 DAC 用于数据完整性服务，为近 20 个不同的应用系统提供服务，日均使用次数约 2000 万次，其关键代码如下：

DAC 服务封装

```
<!-- Activation Service portType Declaration -->
<wsdl:portType name="ActivationCoordinatorPortType">
  <wsdl:operation name="CreateCoordinationContext">
    <wsdl:input  message="wscoor:CreateCoordinationContext"/>
     <wsdl:service name="Data_Dac">    // 服务名
      <wsdl:service path="libencrytp.crypt.data.dac"/>    // 服务目录
        </wsdl:operation>
</wsdl:portType>
<!-- Activation Requester portType Declaration -->
<wsdl:portType name="ActivationRequesterPortType">
  <wsdl:operation  name="CreateCoordinationContextResponse">
    <wsdl:input  message="wscoor:CreateCoordinationContextResponse"/>
  </wsdl:operation>
  <wsdl:operation name="Error">
      <wsdl:input message="wscoor:Error"/>
      <wsdl:service name="erroutlog">
        <wsdl:service path="liblog.errlog"/>      // 服务异常处理
  </wsdl:operation>
</wsdl:portType>
```

JNI 服务封装

```
#include <jni.h>
#include <assert.h>
jstring stoJstring(JNIEnv* env, const char* pat)
{    jclass strClass = (*env)->FindClass(env,"Ljava/lang/String;");
  jmethodID  ctorID  =  (*env)->GetMethodID(env,strClass,  "<init>",
"([BLjava/lang/String;)V");
  jbyteArray bytes = (*env)->NewByteArray(env,strlen(pat));
  (*env)->SetByteArrayRegion(env,bytes, 0, strlen(pat), (jbyte*)pat);
  jstring encoding = (*env)->NewStringUTF(env,"utf-8");
  return (jstring)(*env)->NewObject(env,strClass, ctorID, bytes, encoding);
}
```

DAC 生成

```
shm_k=( ftok ( profile, 1024);    // 创建 IPC 唯一关键字
shmid = shmget ( shm_k, KEY_SIZE*TR_SIZE+HEAD_SIZE, op-
perm );  //创建共享内存区
shmaddr = ( char * ) shmat ( shmid, ( char * ) 0, 0 );  //将共享内存区映
射到进程空间
offset = shmaddr + HEAD_SIZE + i*TR_SIZE;
memcpy ( offset, &tkey, TR_SIZE );
Strcpy(  POkey,  （char * (readshmmk(  ) );    // 从共享内存中获得
发行密钥
ROkey=_DES_3( Pkey )
POkey_bcd=Exhange_Bcd( Rkey );
_bcd( Dac,S_dac );      //  生成 BCD 信息
```

6 小结

基于 SOA 的银行信息安全系统通过分级密钥管理的方法建立了完善的密钥约束体系，总行、省行、各营业机构互相依赖又互相制约，提高了信息安全的强度，降低了密钥泄露带来的风险。通过 SOA 技术，有效的解决了异构环境下不同业务系统个性化的信息安全需求与统一管理带来的矛盾，使系统既有较强的实用性又具有较强的可管理性。该系统在 2008 年已正式投入使用，系统运行稳定，能够支持千万笔/天的高负载业务访问量，可以较好地满足当前和未来业务发展的需要，系统所采用的技术方案对类似系统的开发具有参考价值。

References (参考文献)

[1] ChenJing . The development and trend of informatiza -tion in bank industry [J]. informatization in China, 2008，24 （3）：20-24
陈静.银行信息化发展趋势 [J]. 中国信息化， 2008，24（3）：20-24

[2] CuiYingan, ChenHao .Information Security System of Commercial Bank in Data Concentration Environment [J]. Computer engineering , 2008 34 (22): 162-164
崔颖安,陈皓 .大集中环境下商业银行信息安全系统的研究 [J]. 计算机工程, 2008 34 (22): 162-164

[3] Drik Krafzig, Karl Banke, Dirk Slama. Enterprise SOA Best Practice., 2006

[4] Thomas Erl. Service-Oriented Architecture Concepts, Technology, and Design 2006

[5] H.M.Deitel B.DuWaldt Web Services A Technical Introduction 2006

[6] ISO/IEC 11770-2 information technology-security technology-key management the second part：adopt the mechanism of symmetric technology

[7] ISO/IEC JTC1/SC27 The safety tecjnology standards of information technology

[8] ISO/IEC 9798-2:1994 information technology-security technology-entity identification mechanism-the second part： adopt the entity identification of symmetric encryption algorithm

[9] ISO 8372:1987 information process- operation mode Of 64bit field encryption algorithm

[10] ISO/IEC 10118-1:1994 information technology-security technology-hash function ,the first part：summary

[11] LiJianhua, ChenSongqiao and MaHua. The reference model and application research of SOA by [J].computer engineering, 2006, 32(20):100-101
李建华, 陈松，马华. SOA 参考模型与应用研究 [J].计算机工程, 2006, 32(20):100-101

[12] Enhance the Dependability of Computing Systems: Integration of Virtualization and SOA [J] software journal ，2008,19(5): 1224-1233
宋军,李卫. 基于SOA的动态反射式软件体系结构的研究 [J] 软件学报，2008,19(5): 1224-1233

[13] W.Richand Sieresl.Advanced Programming in the unix Enviornment. 2006.