中国光大银行

网上支付对帐技术规范

1.2 版

目录

前言

第1章　文档概述

1.1 介绍

1.1.1 概述

1.1.2 目标读者

本文的主要目标读者是商户的技术实施人员。

1.1.3 版本规范

1.1.4 最近修订

版本号	作者	内容提要	核准人	发布日期
1.0	金其林	初始创建	徐华山	2012-06-29
1.2	贾云	对账文件增加金额格式	范春	2014-11-04

1.2 参考标准与文献

第2章　业务概述

本章介绍网上支付对帐功能的业务模型，描述对帐功能的主要业务目的，以及各个参与方的功能与责任。

2.1 业务模型

2.2 核心业务

第3章 系统架构

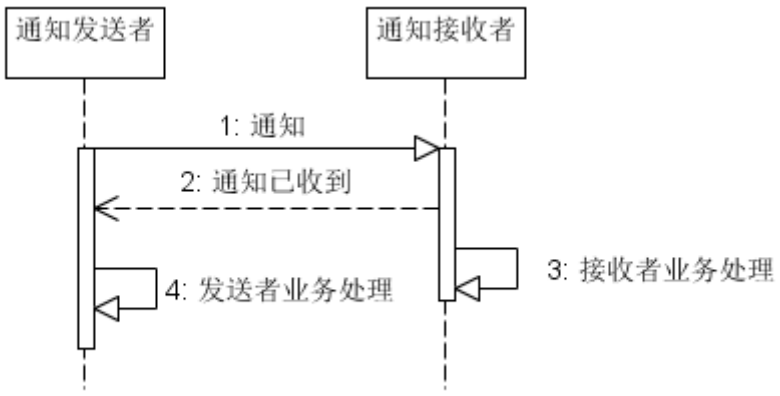
3.1 交互模式

在接口对帐业务的技术实现中，商户与银行之间通过交换报文来交换业务信息、实现业务流程、控制业务规则。

实现接口对帐业务所需的交互模式为请求-应答模式。

3.1.1 单向通知模式

在单向通知模式下，一方作为通知发送者，一方作为通知接收者。发送者发送通知，并保证接收者收到通知。通知接收者在收到通知之后，立即返回发送者通知已收到。通知送达之后，发送方与接收方可以独立地进行后续业务处理。

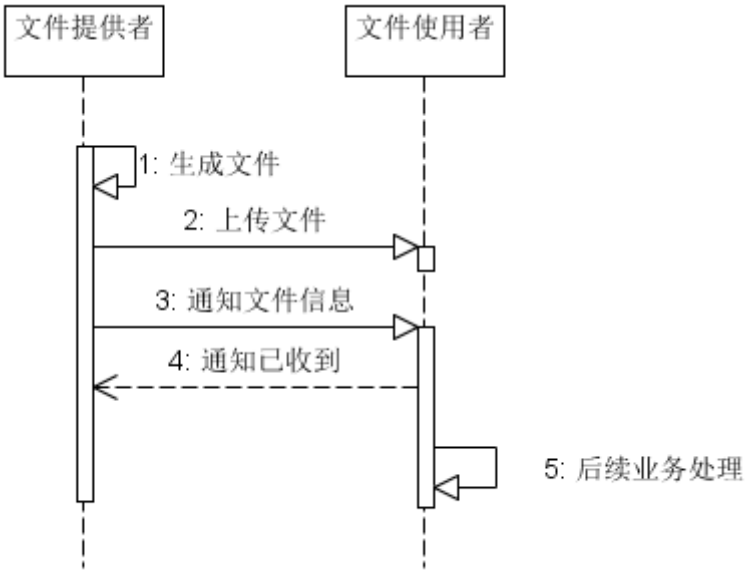


3.1.2 文件上传模式

在文件上传模式中，一方作为文件提供者，另一方作为文件使用者。文件提供者首先生成文件，然后将文件上传给文件提供者。在文件上传完成之后，文件提供者通知文件使用者，通知信息中包含文件的上传位置与其它信息。文件使用者接到通知之后，根据文件内容进行后续的业务处理。

文件上传模式的特点是文件使用者拥有文件服务系统。由于接口对账架构中，文件服务系统是由商户统一提供的，因此，文件上传模式适用于由银行向商户发起的批量业务处理请

求，如清算对账等。



第4章　业务实现规范

网上支付对帐业务功能包含网上支付、退货等交易清算对账功能，商户需要按照技术规范进行开发接入，清算对账功能分为通过企业网银手动下载实现和接口程序接口自动上传实现两种满足不同商户需求。

本章描述商户与银行之间的业务流程与规则。同时，本章也规定了在商户与银行网上支付对帐业务间的交互模式、交换的报文规范。

4.1 接口实现说明

序号	接口类型	实现方式
1	清算对账 一	企业网银（已实现）
2	清算对账 二	接口对帐（选做）

4.2 清算对账 模式一（手动下载）

4.2.1 业务功能

商户发起的网上支付、退货交易的执行结果均以银行端交易状态为准。当发生网络丢失报文或系统故障等情况时，可能会造成银行返回给商户的交易应答的丢失，造成客户银行卡账户中的资金变动与商户账户中的资金变动不一致的状况。除交易应答丢失产生清算结果不一致的情况外，还可能因其它原因（例如系统中的缺陷，或者人为因素），造成清算结果不一致。清算对账业务的目的，是发现用户银行卡账户与商户账户中的资金变动不一致的情况，确定原因，并找出处理方案。

清算对账业务通常每日进行一次，由银行在 T 日日终核对完成之后，生成商户在 T 日所有涉及到资金变动交易明细的数据文件，商户依据此文件对 T 日的交易数据进行比对，并找出不一致的交易记录。对于由银行交易应答丢失造成的不一致的交易记录，由商户进行恢复处理。对于由其它原因造成的清算不一致的情况，双方沟通找出原因，根据确认的原因确定资金处理差错的一方，并由该方进行账务方面的调整处理。

4.2.2 业务规则

清算对账业务在执行中需要满足以下约束条件：

- 银行提供的 T 日交易明细数据与银行的账务处理一致
- 银行在 T+1 日为商户提供 T 日的交易明细数据文件。
- 商户记录的 T 日交易净额与银行清算给商户的 T 日款项净额相同。
- 当清算对账完成，并对差错处理之后，双方 T 日及之前的所有交易引起的资金变动须保持一致。

4.2.3 处理流程

清算对账的处理流程如下：

1. 银行在 T 日日终处理时，将 T 日商户发起的支付和退货交易生成统一的清算对账文件并保存，其中所有的交易均为：经与银行账务系统核对、交易状态为“成功”的记录。
2. 商户需登陆企业网银自行下载清算对账文件，并从对账文件中根据我方提供的格式，解析出每一条交易记录，与记录的在该清算日期的所有交易记录进行逐笔核对。找出并生成以下三类交易记录的清单。
   - A. 银行记录中有，但商户没有的交易记录
   - B. 商户记录中有，但银行没有的交易记录
   - C. 双方记录中都有，但内容不一致的交易记录

商户对不一致的交易记录进行恢复与处理。对于A类存在差别的交易记录，商户进行交易恢复，使商户对该笔交易的资金处理与银行一致。对于B类与C类存在差别的交易记录，商户人工介入处理：通过核对原始的银行应答指令，找出出现差错的原因并确定该由哪方进行调整。如果存在需要银行进行调整的交易记录，则商户线下提供给银行的需要调整的交易记录清单，以及原始的银行应答指令，由银行在人工核实之后进行处理。

4.2.4 交互模式

在清算对账业务中，商户需要登陆光大银行企业网银，自行下载每日的清算对账文件。

企业网银下载对账文件的菜单操作流程为：

商户管理 一级菜单 –> 对账文件 二级菜单。

4.2.5 文件格式

■ **“清算对账”文件结构**

对账文件的内容是以“|”分隔的如下内容，每条记录以回车换行结束

具体记录内容格式如下：

交易代码丨清算日期丨交易发生时间丨订单号丨网关流水号丨商户号丨终端号丨交易金额丨手续费丨净清算金额丨响应码丨商户保留 1丨商户保留 2丨

其中，交易代码的定义如下：

1) ZF01 支付
2) ZF02 退货

注：金额格式——1 元就显示 1，不会显示 1.0 或 1.00。

4.3 清算对帐 模式二（自动上传）

4.3.1 业务功能

商户发起的网上支付、退货交易的执行结果均以银行端交易状态为准。当发生网络丢失报文或系统故障等情况时，可能会造成银行返回给商户的交易应答的丢失，造成客户银行卡账户中的资金变动与商户账户中的资金变动不一致的状况。除交易应答丢失产生清算结果不一致的情况外，还可能因其它原因（例如系统中的缺陷，或者人为因素），造成清算结果不一致。清算对账业务的目的，是发现用户银行卡账户与商户账户中的资金变动不一致的情况，确定原因，并找出处理方案。

清算对账业务通常每日进行一次，由银行在 T 日日终核对完成之后，生成商户在 T 日所有涉及到资金变动交易明细的数据文件，商户依据此文件对 T 日的交易数据进行比对，并找出不一致的交易记录。对于由银行交易应答丢失造成的不一致的交易记录，由商户进行恢复处理。对于由其它原因造成的清算不一致的情况，双方沟通找出原因，根据确认的原因确定资金处理差错的一方，并由该方进行账务方面的调整处理。

4.3.2 业务规则

清算对账业务在执行中需要满足以下约束条件：

- 银行需要在 T+1 日上午 6:00 点前主动向商户提供 T 日协议交易明细数据文件。
- 商户记录的 T 日交易净额与银行清算给商户的 T 日款项净额相同。
- 当清算对账完成，并对差错处理之后，双方 T 日及之前的所有交易引起的资金变动须保持一致。

4.3.3 处理流程

清算对账的处理流程如下：

1、 银行在 T 日日终时，将 T 日与银行账务系统核对正确的支付、退货（包括单笔与批量）

的交易记录生成清算对账文件并保存。不同交易性质的交易保存在同一个清算对账文件中。

2、 银行按照事先约定的上传 URL 与文件名格式向商户上传清算对账文件。（参考第 6 章）

3、 如果商户采用 HTTP 方式接收文件，接收完文件流后通过**“文件上传已被接受”报文 FileAccept** 通知银行文件已传送成功。如果商户采用 FTP 方式接收文件，则不需要返回“文件上传已被接受”报文 FileAccept。

4、 HTTP 方式传输时，银行端收到“文件上传已被接受”报文 FileAccept 或 FTP 方式传输时，银行端直接将文件上传到商户 FTP 服务器的指定目录后，以清算对账日期、文件名等信息构造**“清算对账”通知报文 MCNotify**，告知商户清算对账文件已上传。

5、 商户收到“清算对账”通知报文后，通过**“单向通知已被接受”报文 NotifyAccept** 通知银行。

6、 商户从银行提供的清算对账文件中根据我方提供的格式，解析出每一条交易记录，与商户记录的在该清算日期的所有交易记录进行逐笔核对。找出并生成以下三类交易记录的清单。

- A. 银行端记录中有，但商户没有的交易记录
- B. 商户记录中有，但银行没有的交易记录
- C. 双方记录中都有，但内容不一致的交易记录

7、 商户对不一致的交易记录进行恢复与处理。对于 A 类存在差别的交易记录，商户进行交易恢复，使商户对该笔交易的资金处理与银行一致。对于 B 类与 C 类存在差别的交易记录，商户人工介入处理：通过核对原始的银行应答指令，找出出现差错的原因并确定该由哪方进行调整。如果存在需要银行进行调整的交易记录，则商户线下提供给银行的需要调整的交易记录清单，以及原始的银行应答指令，由银行在人工核实之后进行处理。

4.3.4 交互模式

在清算对账业务中，银行与商户通过文件上传模式上传文件，通过单向通知模式进行交互。

4.3.5 报文格式

- **“接口对账”通知报文 MCNotify(Merchant Clearing Notify)**

清算对账通知报文是银行向商户发起的清算对账通知请求。

中文域名	字段名	类型	是否必输	说明
交易代码	transId	char (10)	是	**MCNotify**
商户号	merId	char (12)	是	
流水号	serialNo	char(19)	是	
交易日期和时间	date	char(17)	是	YYYYMMDD HH:MM:SS
文件 URL	fileURL	char(30)	否	
文件名称	fileName	char(30)	是	lg370310000004201203 03.zip （lg+12 位商户号+8 位日期）。
对账日期	clearingDate	char(8)	是	YYYYMMDD

文件摘要	digest	char(256)	是	

4.3.6 文件格式

■ **“清算对账”文件结构**

对账文件的内容是以“|”分隔的如下内容，每条记录记录以回车换行结束

具体记录内容格式如下：

交易代码丨清算日期丨交易发生时间丨订单号丨网关流水号丨商户号丨终端号丨交易金额丨手续费丨净清算金额丨响应码丨商户保留 1丨商户保留 2丨

其中，交易代码的定义如下：

1) ZF01 支付
2) ZF02 退货

注：金额格式——1 元就显示 1，不会显示 1.0 或 1.00。

第5章 报文规范

报文规范是网上支付接口对帐技术标准中重要的组成部分，规定了商户与银行之间交换报文的顺序、格式、语义与处理规范。本章中介绍报文的一般结构与公共元素，这些规范适用于所有的接口对帐报文。具体业务中使用的报文，在各个业务实现规范中分别进行阐述。

5.1 报文结构

网上支付接口对帐报文统一采用 xml 格式（见附录）。所有的接口对帐报文均以 MessageSuit 作为根元素，每个 MessageSuit 元素中可以包含多个 Message 元素。Message 元素中包含代表具体数据的元素，比如 Plain、Signature 等。每个数据元素由一系列属性元素构成。

作为约定，MessageSuit 元素、Message 元素与业务元素均是首字母大写的 CamelCase 形式，所有的属性元素均是首字母小写的 CamelCase 形式。

以单向通知已被接受报文为例，签名前准备数据的格式如下：

```
<MessageSuit><Message id="..."><Plain id="..."><transId>…</transId ><merId>…</merId>
</Plain></Message></ MessageSuit>
```

签名后报文的格式如下：

```
< MessageSuit>
  <Message id="...">
    < Plain id="...">
        <transId>...</transId>
        <merId>...</merId>
    </ Plain>
    <Signature>…</Signature>
```

```
  </Message>
</ MessageSuit>
```

5.2 报文分类

网上支付接口对帐中的报文按照交互模式的不同，分为以下几类：

- 服务请求类报文

服务请求类报文用于请求-应答交互模式，由服务使用者向服务提供者发送。服务请求类报文的命令规范是 XXNotify，其中 XX 是报文代表的业务的首字母缩略，Req 是 Request 的缩写。比如对于清算对账通知报文，命名为 MCNotify，代表 **M**erchant **C**learing **N**otify。

- 服务应答类报文

服务应答类报文用于请求-应答交互模式，由服务提供者向服务使用者返回。服务应答类报文的命令规范是 XXAccept，其中 XX 是报文代表的业务的首字母缩略，Res 是 Response 的缩写。比如对于单向通知已被接受报文，命名为 NotifyAccept。

- 通用报文

通用报文适用于所有的接口对帐业务。接口对帐中的通用报文为 Error 报文，用于当请求不能被正确处理时的应答返回。

对于和业务相关的服务请求类、服务应答类报文的具体格式，将放在具体的业务实现规范中加以描述。通用报文将在本章描述。

5.3 通用报文

5.3.1 错误消息报文 Error

- **功能**

用来当请求不能被正确处理时应答返回

- **Error 域**

下表列举了 Error 消息域的定义

中文域名	字段名	类型	是否必输	说明
交易代码	transId	char (10)	是	Error
商户号	merId	char (12)	是	
错误代码	errorCode	char(4)	是	
错误描述	errorMessage	char(256)	是	
详细错误信息	errorDetail	char(512)	是	

- **错误代码说明**

下表列举了标准的错误代码：

错误代码	错误描述	解释
程序类错误		
0002	必填域缺失	接口中必输项没有送值
0125	账户类型错误	银行卡的类型不支持此项业务
0212	商户校验失败	该商户不支持网上支付业务/该商户的服务类型不是快捷支付
0400	支付流水重复	重复的网上支付流水
用户类错误（可以将错误信息显示给用户）		
1206	认证信息不匹配	认证信息与商户通过认证的信息不匹配
1601	金额超限	支付金额超过每日限额
1602	余额不足	银行账户中的余额不足以完成支付
1605	银行交易失败	该笔交易在银行系统中已经失败，且状态不会再发生变更。错误具体详情通过 ErrorDetail 告知
管理类错误		
2000	接口对帐渠道关闭	没有开通接口对帐业务
系统类错误		
9000	暂时系统异常	例如，系统参数正在处理中。

5.3.2 NotifyAccept 报文

■ **功能**

用来代表单向通知已被接受。

■ **消息域**

下表列举了消息域的定义

中文域名	字段名	类型	是否必输	说明
交易代码	transId	char (10)	是	NotifyAccept
商户号	merId	char (12)	是	

5.3.3 FileAccept 报文

■ **功能**

用来代表文件上传已被接受。

■ **消息域**

下表列举了消息域的定义

中文域名	字段名	类型	是否必输	说明

交易代码	transId	char (10)	是	FileAccept
商户号	merId	char (12)	是	

5.4 报文的解析与传输

接口对帐报文的传输使用 XML Over HTTPS 方式，在 HTTP 请求/响应体中包含 XML 形式的报文。

5.4.1 报文解析

对 XML 解析的基本要求如下：

- **xml 解析**

为了可以支持后续快捷版本，xml 解析的实现不要做严格的验证。特别是需要忽略未被确认的域。所有 xml 消息必须用“UTF-8”编码。

- **Message 域之 id 属性匹配**

请求和应答报文的 **Message** 域之 id 属性必须相同，id 是请求方生成的唯一序列号。比如：商户在 USCReq 的 Message 域设置了一个 id 属性值，则银行在 USCRes 里面的 Message 域的 id 属性则与 USCReq 的 Message 域之 id 值相同。

5.4.2 报文传输

对 HTTPS 传输的基本要求如下：

- **使用 POST 发送消息**

消息请求基于 HTTPS 的 POST 方式。

- **HTTP 消息头要求**

HTTP请求与响应消息中必须按照如下要求设置头部域：
‘**Content-Length:**’必须设置成消息体的长度
‘**Content-Type:**’必须设置下面的值：**application/xml; charset=utf-8**

第6章 文件规范

6.1 文件命名规范

文件命名规范对文件名称进行统一的规划，以达到从文件名称上区分不同业务文件的目的。文件命名规范：lg+merchantno+yyyymmdd.zip，其中：

- lg：固定
- merchantno：商户号
- yyyymmdd：文件业务日期（如：清算日期）。

6.2 文件交换规范

双方交互的文件存储在商户端的文件服务器上，当采用 HTTP 方式传输时，访问 URL 格式如下：
https://MerchantFileUrl/action/instId/yyyymmdd/filename?certId=xxxx&sign=xxxxxx ， 其中：

- ✧https:// MerchantFileUrl /——商户文件系统的根路径（域名由商户确定）。
- ✧action——操作类型（如：upload 表示上传、download 表示下载）。
- ✧instId——银行机构代码（CEB）。
- ✧yyyymmdd——文件业务日期（银行清算日期）。
- ✧filename——遵循业务文件命名规范的文件名。
- ✧sign——使用商户的公钥对字符串“/action/instId/yyyymmdd/filename”采用 SHA1WithRSA 算法进行签名，对签名结果进行 Base64 编码获得的字符串。

注：文件上传时，文件通过附件方式上传。如果文件上传成功，服务器返回 FileAccept 报文，如果文件上传失败，服务器返回 Error 报文。

6.3 文件压缩

传输前需要压缩成 zip 格式。

6.4 文件签名

对压缩后的文件进行数字签名，签名算法使用标准 MD5withRSA 算法。

签名信息不包含在文件中，通过文件上传通知报文传送(参考“接口对账”通知报文中，digest 字段)。文件上传完成后，文件上传者发送文件上传通知报文将签名传送给文件使用方。

6.5 文件上传规则

文件上传完成后，文件上传者发送文件上传通知报文给对方。

第7章 信息安全规范

本章主要从技术角度阐述快捷系统中可能出现的安全威胁，以及通过技术规范来规避安全风险的方式。

7.1 安全威胁

接口对帐业务涉及到商户与银行系统通过公众互联网络交换信息与指令。因此，在接口对帐信息安全规范中，我们主要对来自网络的安全威胁进行分析。对于来自内部系统与人员的安全风险，由银行与商户现有的安全体系来保障。

来自网络的安全威胁主要有以下几种：

- 交易指令篡改

如果网络中传输的快捷交易指令被入侵者截获并篡改，就会造成资金处理出现错误，给接口对帐业务参与方造成损失。因此，在网络传输中，需要保证指令完整性。

解决方案：使用数字证书对对报文中的业务数据进行签名（见数字签名）。

- 交易指令伪造

如果有入侵者冒充商户或者银行发起交易指令，就会造成资金在未得到接口对帐业务参与方的授权下发生流动，给银行、商户或客户带来损失。因此，在网络传输与系统处理中，需要保证指令的真实性。

解决方案：使用数字证书对对报文中的业务数据进行签名（见数字签名）。

- 交易指令否认

当发生交易纠纷时，双方需要通过交易指令确定责任方。接口对帐安全规范中需要为交易指令防否认提供技术支持。

解决方案：使用数字证书对对报文中的业务数据进行签名（见数字签名），银行与商户均应保留涉及资金变动的交易报文日志（见报文日志管理）。

- 交易指令重播

入侵者也可能尝试通过截获网络中传输的接口对帐交易指令，并多次重播的方式试图发起未经授权的交易。在接口对帐技术规范中，需要防止交易指令重播引起的未授权交易。

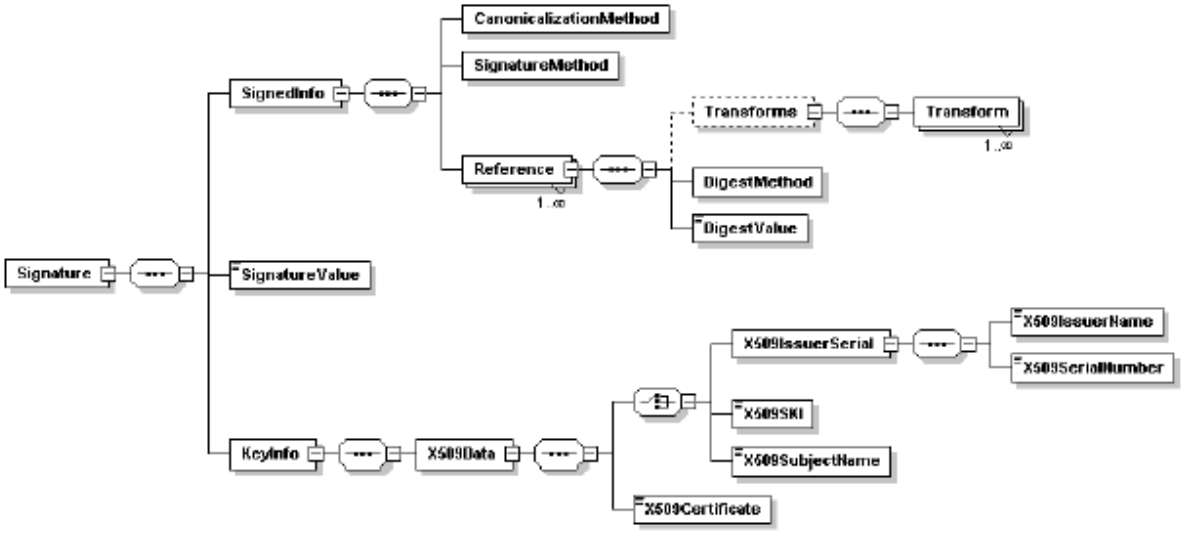
解决方案：凡涉及到资金变动的交易指令，流水号必须唯一。银行与商户均需要保证同一流水号的交易指令只能执行一次。

7.2 数字签名

7.2.1 数字签名要求

必须采用本节所述方法对 MessageSuit 消息进行数字签名。签名规范遵循 XML-Signature Syntax and Processing, W3C Recommendation 规范（http://www.w3.org/TR/xmldsig-core/）。

MessageSuit 协议使用分离签名（Detached signature），即<Signature> 元素与被签名的元素各自独立存在。被签名的元素和 <Signature> 元素包含在同一文档中。签名元素通过当地引用（如'# Plain1234'）被引用。被签名的元素内容包括从 Plain 等开始标签的开始括号开始到 Plain 等结束标签的结束括号为止的内容。



签名结构图

MessageSuit 签名的产生必须满足下列表格中定义的元素内容和算法要求。

表1 XML Signature Profile

元素	要求
Signature	没有KeyInfo实例；没有Object实例
CanonicalizationMethod	元素为空，但出现Algorithm属性
SignatureMethod	元素为空，但出现Algorithm属性
Transforms	存在且都包含一个Transform的实例
Transform	元素为空，但出现Algorithm属性
DigestMethod	元素为空，但出现Algorithm属性
KeyInfo	不出现。

表2 XML 签名算法

算法类型	标识符
Canonicalization	http://www.w3.org/TR/2001/REC-xml-c14n-20010315
Digest	http://www.w3.org/2000/09/xmldsig#sha1
Encoding	http://www.w3.org/2000/09/xmldsig#base64
Signature	http://www.w3.org/2000/09/xmldsig#rsa-sha1

Transform	http://www.w3.org/2000/09/xmldsig#enveloped-signature

7.2.2 规范化要求

注意：规范化是 “XML-Signature Syntax and Processing, W3C Recommendation”中的要求之一，也叫作xmldsig。

Xmldsig 表示计算同样文档的摘要必须使用规范化的方法。

7.2.3 签名的 XML 命名空间

消息的签名必须被声明在一个缺省的命名空间：http://www.w3.org/2000/09/xmldsig#中。

示例：

```
< MessageSuit>
    <Message id="901239052203">
        <Plain id="NotifyAccept">
            <transId>NotifyAccept</transId>
            <merId>370310000004</merId>
        </ Plain>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
            <SignedInfo>
                <CanonicalizationMethod
                    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315">
                    </CanonicalizationMethod>
                <SignatureMethod
                    Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1">
                    </SignatureMethod>
                <Reference URI="#TSReq">
                    <Transforms>
                        <Transform
                            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature">
                            </Transform>
                    </Transforms>
                    <DigestMethod
                        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
                        </DigestMethod>
                    <DigestValue>4t79dXZ7/BQqgiBdkziaKTUslVU=</DigestValue>
                </Reference>
            </SignedInfo>
            <SignatureValue>

    RyWMXvxlhmLuOIOvGSIkpV1iRV400F1B2W0zmW9nc5qfvfNMtpyp+uNwksBUjcO8H//NW4GvxcDd
```

```
                    KMuuc6k5MDMH1E4OUg+624FUY23qs3R2ztubtD3MU7xk4f0iq9L16GK4ZBeID/Lyj6CxjaCcp3Fu
                    K1CznNj4Kr+qRLtxx+s= </SignatureValue>
            </Signature>
        </Message>
</ MessageSuit >
```

7.3 软件包接口说明

7.3.1 XML 签名

String
com.csii.payment.client.core.MerchantXmlSignVerify.merchantSignXmlData_ABA(String PlainData, String TransId);
功能描述：静态方法，用于商户使用自己的私钥对 XML 签名格式的原始数据进行数字签名
输入参数：签名的原始字符串 PlainData
输入参数：签名数据的交易码 TransId
输出参数：签名的目标字符串 SignData

7.3.2 XML 验签

boolean
com.csii.payment.client.core.MerchantXmlSignVerify.merchantVerifyXmlData_ABA(Sign)
功能描述：静态方法，用于商户使用对银行传递来 XML 签名格式的信息进行校验
输入参数： 签名的目标字符串 SignData
输出参数： true - 校验成功 false - 校验失败

7.4 签名及验签的调用

签名是在商户拼装好交易字段的明文后，调用商户的证书私钥进行签名的过程。网关将校验该签名以保证该交易是该商户发起。通常，在商户通过https页面方式或者https指令的方式给网关发起交易的时候，都必须对其交易字段明文进行签名。

验签是网关处理完交易（比如支付、查询等）后，组织好交易结果的明文后，调用网关证书私钥对明文进行签名，商户将通过校验该签名以保证结果是银行网关返回的。通常，在商户接收各种交易结果的时候，都必须对该签名字段进行验签。

以下章节以https页面方式下的发送订单和支付后的交易结果发送为例，分别描述签名及验签的用法。

7.4.1 XML 签名方法调用

以下内容以HttpClient方式发送订单交易为例，演示签名方法的调用。

示例如下：

在程序中调用方法

```
com.csii.payment.client.core.MerchantXmlSignVerify.merchantSignXmlData_ABA( String PlainData, String TransId );
```

方法，将得到数字签名通过SignData字段传递给银行。

```
String bankUrl = “https://xx.xx.xx.xx/payment/ceb_pay.do”;
HttpClient httpclient = new HttpClient();
PostMethod post = new PostMetchod(bankUrl);
post.setRequestEntity(new ByteArrayRequestEntity(SignData.getBytes()));
result = httpClient.excuteMethod(post);
```

7.4.2 XML 验签方法调用

从网关返回的信息中，获得银行的签名数据，

```
Byte[] signBytes = post.getResponseBody();
String signData = new String(signBytes,"UTF-8");
```

然后调用方法：

```
com.csii.payment.client.core.MerchantXmlSignVerify.merchantVerifyXmlData_ABA (String SignData)
```

，校验签名。

7.5 应用部署

- 首先，将商户密钥容器文件cebmerchant.jks放在ceb_merchant.properties中cafile指示的文件系统位置。
- 然后，按照J2EE WEB应用的部署规范：
  - 将修改后的ceb_merchant.properties文件放置到/WEB-INF/classes目录
  - 将cebmerchant.jar、cebmerchant_ext.jar软件包放在/WEB-INF/lib目录下

将 cebmerchant.jks 文件放在 ceb_merchant.properties 文件中指定的地方。

7.6 报文日志管理

为了做到交易指令防否认，银行与商户均应保留涉及资金变动的交易报文日志，日志中包括完整的报文、报文签名、接收时间，保存时间不小于 90 天。

交易报文日志应该妥善管理，避免泄密或者被破坏。

第8章 其它规范

8.1 异常处理规范

凡是报文验证不通过，或者业务处理失败，统一返回通用错误报文 Error。通用错误报文 Error 的格式参见 5.3.1 节。

8.2 错误码规范

错误码包含两部分，一部分是标准错误码，由接口对帐标准规定，参见 5.3.1 节。另一部分是商户特定代码，由商户自行规定。

第9章 附录

9.1 报文格式 DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT MessageSuit(Message)+>
<!ELEMENT Message (Plain, Signature) >
<!ATTLIST Message id ID #REQUIRED>

<!ELEMENT Plain (version,transId,merId,relatedAcct?,date?,name?,cardNo?,cardType?,validDate?,cvv2?,certType?,certNo?,phone?,bussType?,stageFlag?,stages?,serialNo?,charge?,amount?,currency?,goods?,subMerName?,clearDate?,type?,originalSerialNo?,originalDate?,status?,errorCode?,errorMessage?,errorDetail?,Signature)>
<!ATTLIST Plain id NMTOKEN #REQUIRED>
<!-- 属性 -->
<!ELEMENT version (#PCDATA)>
<!ELEMENT transId (#PCDATA)>
<!ELEMENT merId (#PCDATA)>
<!ELEMENT relatedAcct (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT cardNo (#PCDATA)>
<!ELEMENT cardType (#PCDATA)>
<!ELEMENT validDate (#PCDATA)>
<!ELEMENT cvv2 (#PCDATA)>
<!ELEMENT certType (#PCDATA)>
<!ELEMENT certNo (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

```
<!ELEMENT bussType (#PCDATA)>
<!ELEMENT stageFlag (#PCDATA)>
<!ELEMENT stages (#PCDATA)>
<!ELEMENT serialNo (#PCDATA)>
<!ELEMENT charge (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT goods (#PCDATA)>
<!ELEMENT subMerName (#PCDATA)>
<!ELEMENT clearDate (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT originalSerialNo (#PCDATA)>
<!ELEMENT originalDate (#PCDATA)>
<!ELEMENT status (#PCDATA)>
<!ELEMENT errorCode (#PCDATA)>
<!ELEMENT errorMessage (#PCDATA)>
<!ELEMENT errorDetail (#PCDATA)>
<!ELEMENT Signature (#PCDATA)>
```

9.2 金额格式

金额以元为单位，保留小数点后 2 位，最多 12 位。例如：如果交易金额为 123.45，即表示一百二十三元四角五分。

9.3 货币代码表

接口对帐报文中的货币代码（currency 元素）的取值遵照国家标准 GB/T 12406-1996《表示货币和资金的代码》，该标准中规定了代表货币和资金的三个字母代码和与之等价的三位数字代码的结构，说明了货币单位与货币之间十进制的关系，确立了维护代理机构的建立过程，并详细说明了代码应用方法。

根据 GB/T 12406-1996，货币代码字段使用 3 位定长数字，如下表所示：

国家、地区名称	货币名称	货币代码
中国	人民币元	156
中国香港	港元	344
中国澳门	澳门元	446
美国	美元	840
英国	英镑	826
法国	法国法郎	250
德国	马克	278
俄罗斯	卢布	810
日本	日元	392

加拿大	加元	124
瑞士	瑞士法郎	756
瑞典	瑞典克朗	752
意大利	意大利里拉	380
西班牙	西班牙比塞塔	724
葡萄牙	葡萄牙埃斯库多	620
荷兰	荷兰盾	528
比利时	比利时法郎	056
芬兰	马克	246
挪威	挪威克朗	578
希腊	德拉克马	300
奥地利	先令	040
丹麦	丹麦克朗	208
澳大利亚	澳大利亚元	036
新西兰	新西兰元	554
巴西	克鲁赛罗	076
南非	兰特	710
埃及	埃及镑	818
伊拉克	伊拉克第纳尔	368
伊朗	伊朗里亚尔	364
沙特阿拉伯	沙特里亚尔	682
科威特	科威特第纳尔	414
阿联酋	UAE 迪拉姆	784
泰国	铢	764
新加坡	新加坡元	702
印尼	卢比	360
马来西亚	马来西亚林吉特	458
菲律宾	菲律宾比索	608
南朝鲜	圆	410
印度	印度卢比	356