手机网站支付开发指南 PHP

文件版本：3.0.0

2012-07-24

版本号	作者	内容提要	核准人	发布日期
0.1	胡叶军(薛刚)	支付宝 wap 开发指南		
0.2	陈枫（炎汐）	新增接口描述、通知部分说明		2009-09-20
0.3	陈枫（炎汐）	修改线上服务器接入地址		2009-11-06
0.4	陈枫（炎汐）	交易创建接口新增 out_user 参数与 zero_pay 参数		2009-11-26
1.0	丁朗、曹腾	内容梳理修订		2010-09-27
1.0.1	曹腾	新增 pay_expire 参数		2010-10-28
1.0.2	曹腾	buyer_account_name 参数屏蔽		2010-12-27
1.0.3	洛心	交易创建接口增加 call_back_url 参数，授权并执行接口删除此参数， 签名注意事项修改		2011-04-20
1.0.4	洛心	内容修订，增加 UC 浏览器		2011-06-29
1.0.5	洛心	修改 submit_img_url 参数		2011-07-25
1.0.6	洛心	整合支付方式前置		2011-09-06
2.0.0	许宏杰（正太）	整体结构调整和重新编写		2011-11-04
2.0.1	许宏杰（正太）	增加支付前置 sign_type=0001，修改相关开发步骤		2011-11-28
2.0.2	许宏杰（正太）	修改 2.3.2 示例中的 sign 参数		2011-12-29
2.0.3	许宏杰（正太）	修改常见问答和 notify_url 通知		2012-3-5
2.0.4	巍然	修改完善 RSA 密钥配置内容；前置渠道添加 out_user 参数		2012-04-18
3.0.0	许宏杰（正太）	去除支付前置并重新整理文档结构		2012-7-11

前言

1. 面向读者

本文档主要面向需要接入支付宝手机网站支付的商户的开发人员。

2. 读者所需技能

读者需有 PHP 程序开发背景，掌握 PHP 与 Apache 服务器等相关技能。

3. 开发环境要求

本 Demo 在 php 5.2.6 下测试正常，如商户是其他版本，可以自行测试，只要相关扩展库支持即可。

目录

第一章手机网站支付服务简介

1.1 服务介绍

1.1.1 Wap 支付

步骤一 ： 调用接口 alipay.wap.trade.create.direct， 提交订单信息，获取 token 串。

步骤二 ： 调用接口 alipay.wap.auth.authAndExecute，提交 token 串，跳转到支付宝收银台。

步骤三 ： 处理支付宝系统通知。

基于 http/https 的 请求/响应模式。建议使用 http 请求已适配更多机型。

http 请求地址:http://wappaygw.alipay.com/service/rest.htm

https 请求地址:https://wappaygw.alipay.com:443/service/rest.htm

第二章接入流程

2.1 接入前期准备

接入前期准备工作包括**商户签约**和**密钥配置**，已完成商户可略过。

2.1.1 商户签约

首先，商户需要在 https://ms.alipay.com 进行注册，并签约安全支付服务。签约成功后可获取支付宝分配的账户 ID(PID)，账户名(Seller_account_name)，如图：

图 2- 1 商户 ID 获取示意图

签约过程中需要任何帮助请致电：0571-88158090（支付宝商户服务专线）

2.1.2 密钥配置

无线商户与支付宝交互加密一共有 2 种形式（MD5、RSA），RSA 的接入难度比 MD5 高，但是也比 MD5 更安全，防抵赖；两种方式选其一

签约成功后，商户可登录 https://ms.alipay.com 后点击我的商家服务->密钥管理来获取商户账号对应的 MD5 Key 或支付宝公钥。如下图：

至此，接入前期准备工作完成，下一节将使用 demo 测试准备工作是否正确。

2.2 使用 Demo 测试

2.2.1 Demo 配置运行

为了便于商户的接入，我们提供了安全支付 demo。通过本 demo，商户可测试 2.1 节的

前期准备工作是否正确完成，同时还可参考 demo 的代码完成接入。

步骤 1

解压下载的开发资料压缩包 

，点击进去找到 WAP 支付 demo(php 版)文件夹，里面有 2 个加密类型的 Demo，结合商户项目选择一种并导入到您的开发环境中。

步骤 2

打开 Demo 里的 alipay_config.php 文件，把所有留空的参数信息配置完整。

主要参数说明（详细参数列表请看附录 B 手机网站支付接口参数表）

Partner 合作伙伴ID

PrivateKey 商户私钥

Alipaypublick 支付宝公钥

Out_trade_no 外部交易号（每次交易不能重复）

Seller_account_name 商户收款账号（买家在支付完成后即时到账至该账户）

Call_back_url 同步跳转通知页面（买家支付完成后，15秒自动跳转，或点击返回跳转，一般只需要美化界面，告知用户交易状态）

Notify_url 异步跳转通知页面（支付宝发送通知消息给商户服务器的地址，用于商户对该笔订单更新状态等操作，验签通过必须只返回success，不能包含其他文字和任何字符，否则均视为商户返回了fail，请在浏览器源代码中仔细检查）

Merchant_url 取消支付跳转页面（支付过程中点取消返回的页面）

步骤 3

配置完毕，运行本 Demo 程序，会有购买模拟页面，点击购买链接会跳转到支付宝收银台，如果未能跳转，说明您配置的参数信息不正确，请核对您的支付宝卖家信息，务必在 Demo 测试成功之后才能将本 Demo 部分代码集成至您的实际项目中。

本 Demo 代码仅作参考，主要帮助商户能够快速开发，商户技术可以酌情修改。

2.2.2 Demo 结构说明

一些业务逻辑处理类都放在 class 文件夹里，RSA Demo 会多 key 文件夹（用于存放商户私钥、公钥和支付宝公钥）

类文件说明（RSADemo 结构和 MD5Demo 结构是类似的）：

alipay_config.php：该类是配置所有请求参数，接口，商户的基本参数等

alipay_function.php：该类是请求、通知返回和 RSA 解密、签名、验签等方法调用类文件

alipay_service.php：构造支付宝各接口表单 HTML 文本，获取远程 HTTP 数据

alipayto.php：调用创建交易接口和授权并执行接口

call_back_url.php：同步返回通知处理页

Index.php：Demo 模拟首页

Notify_url.php：异步返回通知处理页面，验签成功后必须只能返回 success，失败则返回 fail

2.3 接口集成

2.3.1 创建交易并获取 token

步骤 1

创建待签名字符串，格式例如：

```
format=xml&partner=2088301265823075&req_data=<direct_trade_create_req><subject>请输入商品名称
</subject><out_trade_no>030116396</out_trade_no><total_fee>0.01</total_fee><seller_account_name>xxx@sohu.com</seller_account_name><notify_url>http://www.xxx.com/Notify.aspx</notify_url><out_user>xxxyyyzzz</out_user><merchant_url>http://www.xxx.com/Merchant.aspx</merchant_url><call_back_url>http://www.xxx.com/callback.aspx</call_back_url></direct_trade_create_req>&req_id=030116396&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0
```

字符串以 key、value 连接，多个参数用&分隔，req_data 是 xml 格式字符串，字符串必须按照参数名首字母升序排列，参数含义请看所有参数列表

步骤 2

将以上待签名字符串当做参数调用RSA签名方法，如下：

function sign($data)

参数详解：

$data：待签名数据，也就是上面方框内的字符串

返回值： RSA签名字符串，例如：

eAYqgDoK77/S2GzsHBalc3ezSPui5EO4uxSk1WgHc33Voc3J2DnUjOCCM7/SyIBJI8vuin/1cTZVC5bL9/+uAVBfnnhC+Zk2CeOJwqHnSOOeB29/WZrpPml5lccb9u4cDzWx/fsX8nKwQJb3XYuQFOTdc2misnwIr7KRTKyafos=

RSA签名用到的是商户的私钥，编码是UTF-8。

生成好的签名先URLEncode转码，然后加上sign参数名放入待签名字符串末尾，如下：

```
format=xml&partner=2088301265823075&req_data=<direct_trade_create_req><subject>请输入商品名称</subject><out_trade_no>030116396</out_trade_no><total_fee>0.01</total_fee><seller_account_name>xxx@sohu.com</seller_account_name><notify_url>http://www.xxx.com/Notify.aspx</notify_url><out_user>xxxyyyzzz</out_user><merchant_url>http://www.xxx.com/Merchant.aspx</merchant_url><call_back_url>http://www.xxx.com/callback.aspx</call_back_url></direct_trade_create_req>&req_id=030116396&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0&sign=eAYqgDoK77%2fS2GzsHBalc3ezSPui5EO4uxSk1WgHc33Voc3J2DnUjOCCM7%2fSyIBJI8vuin%2f1cTZVC5bL9%2f%2buAVBfnnhC%2bZk2Ce0JwqHnSO0eB29%2fWZrpPml5lccb9u4cDzWx%2ffsX8nKwQJb3XYuQFOTdc2misnwIr7KRTKyafos%3d
```

步骤3

调用alipay.wap.trade.create.direct接口，POST或者GET方式提交给支付宝网关

http 请求地址:http://wappaygw.alipay.com/service/rest.htm

https 请求地址:https://wappaygw.alipay.com:443/service/rest.htm

GET 方式请求样例：

```
http://wappaygw.alipay.com/service/rest.htm?call_back_url=http://www.xxx.com/Call_Back.aspx&format=xml&partner=2088301265823075&req_data=<direct_trade_create_req><subject>请输入商品名称</subject><out_trade_no>030116396</out_trade_no><total_fee>0.01</total_fee><seller_account_name>xxx@sohu.com</seller_account_name><notify_url>http://www.xxx.com/Notify.aspx</notify_url><out_user>xxxyyyzzz</out_user><merchant_url>http://www.xxx.com/Merchant.aspx</merchant_url></direct_trade_create_req>&req_id=030116396&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0&sign=sign=eAYqgDoK77%2fS2GzsHBalc3ezSPui5EO4uxSk1WgHc33Voc3J2DnUjOCCM7%2fSyIBJI8vuin%2f1cTZVC5bL9%2f%2buAVBfnnhC%2bZk2Ce0JwqHnSO0eB29%2fWZrpPml5lccb9u4cDzWx%2ffsX8nKwQJb3XYuQFOTdc2misnwIr7KRTKyafos%3d
```

POST 方式无法在文档中举例，请商户参考 Demo 实现

步骤 4

商户通过上面步骤请求之后，支付宝会返回数据给商户服务端，返回数据中除了 sign 之外的其他参数都要通过升序排列，然后调用验签方法

1、当商户使用 **RSA** 签名方式时，实际返回的内容如下（其中 res_data 参数值为加密内容（红色部分），商户必须先 URLDecode，然后用商户的 RSA 私钥解密，最后验签）：

```
res_data=Cl2Mm1Z2YILG8oYe8%2FngEAvYSM9YYmcqUqLtUCZ10habqYb6poowofjzVG3nsUJY6qlgnRrq%2FxFTtdLdwBDGltV8rwpf1AFB01ydCanpQoFgQg%2Brt79JRQ%2B9CC3E%2Fg148C4F95eJ1FNf0L6taXaMFwxaxrvTAdDHzzvSigy3%2BaKdFh8z2K1Zs4gm2bD39lR1CRXSipOyVFhCZZR9L9N8tQNZbDqnyBu%2FjLdLbvXvEuE4fImZPPbsALecVCvsHL4iKFrquPnhA4Zz%2FZEM%2FoJghXA6xlAO0a1d0h6Os%2Fd83mvDPfmhs3oVjPX3FsXCL18Dg4mdzj3gWllbqLnwamM94g%3D%3D&service=alipay.wap.trade.create.direct&sec_id=0001&partner=2088201747196380&req_id=12
```

88337908547&sign=RiyyndPEei2QQc%2FHt1%2FIrmYyW6%2FFKNZFxPUiOcXndAOo3OifNRshRjaLlwEs3d2pBpbmyclfooF7tctFdXcrSM584wgsY%2Bj2o0Z6dXst9lmz%2F4OD%2BL2ubk1DXoLWau0f5NiteIuGqGDWUdXMKRLx1FJ0f%2FMN8GOCUZYN15%2FUE%2FE%3D&v=2.0

2、当商户使用**MD5**签名方式时，实际返回的内容如下（其中res_data参数值为明文内容，无需解密，直接验签）

partner=2088101000137799&req_id=1283133204160&res_data=<?xmlversion="1.0" encoding="utf-8"?><direct_trade_create_res><request_token>20100830e8085e3e0868a466b822350ede5886e8</request_token></direct_trade_create_res>&sec_id=MD5&service=alipay.wap.trade.create.direct&v=2.0&sign=72a64fb63f0b54f96b10cefb69319e8a

3、失败返回样例：

失败的 detail 里面会有各种信息，这里的示例是没有开通接口权限的错误，其他错误请商户检查以上步骤是否正确并修改重新提交。

失败返回无论哪种签名方式，内容都是明文无需解密。

partner=208810100013779&req_id=1283133132946&res_error=<?xml version="1.0" encoding="utf-8"?><err><code>0005</code><sub_code>0005</sub_code><msg>partner illegal</msg><detail>合作伙伴没有开通接口访问权限</detail></err>&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0

步骤 5

res_data 参数是 XML 格式的字符串，解析 request_token 节点得到 token 字符串。

<request_token>20100830e8085e3e0868a466b822350ede5886e8</request_token>

2.3.2 授权并执行

步骤 1

创建待签名字符串，格式例如：

format=xml&partner=2088301265823075&req_data=<auth_and_execute_req><request_token>201110259f7686ab763c20e630db9902166f0bfa</request_token></auth_and_execute_req>&sec_id=0001&service=alipay.wap.auth.authAndExecute&v=2.0

字符串以参数名=值表示，多个参数用&分隔，<request_token>里面填上面返回的 token 字符串。

步骤 2

将上面待签名字符串做sign签名，具体方法同上。

生成好的签名先 URLEncode 转码，然后当做 sign 参数拼装到待签名字符串末尾，如下：

```
format=xml&partner=2088301265823075&req_data=<auth_and_execute_req><request_token>201110259f7686ab763c20e630db9902166f0bfa</request_token></auth_and_execute_req>&sec_id=0001&service=alipay.wap.auth.authAndExecute&v=2.0&sign=A1LhhVwoCHT9yVtKKdBLtcwFYbIlAlWO28stm8vuFYwZ%2bcYcT4XMSW5UMVOCbzBZQ76Go04AriB78LPbo%2fAhNO4nxYL%2fJs7rbymQtvVXRGaqtgrMu1JMWpDxUSyoqACPmyusG90vXztXVjzbfquG2BVKfclYcEGOzF1WDiHOMjw%3d
```

步骤 3

调用alipay.wap.auth.authAndExecute接口，浏览器通过GET方式跳转支付宝网关

http 请求地址:http://wappaygw.alipay.com/service/rest.htm

https 请求地址:https://wappaygw.alipay.com:443/service/rest.htm

GET 方式请求样例：

```
http://wappaygw.alipay.com/service/rest.htm?format=xml&partner=2088301265823075&req_data=<auth_and_execute_req><request_token>201110259f7686ab763c20e630db9902166f0bfa</request_token></auth_and_execute_req>&sec_id=0001&service=alipay.wap.auth.authAndExecute&v=2.0&sign=A1LhhVwoCHT9yVtKKdBLtcwFYbIlAlWO28stm8vuFYwZ%2bcYcT4XMSW5UMVOCbzBZQ76Go04AriB78LPbo%2fAhNO4nxYL%2fJs7rbymQtvVXRGaqtgrMu1JMWpDxUSyoqACPmyusG90vXztXVjzbfquG2BVKfclYcEGOzF1WDiHOMjw%3d
```

Header跳转该地址（即：支付宝wap收银台地址）

2.4 处理支付宝系统通知

支付宝系统的通知包括同步和异步两种方式，同步是指在支付完成后支付宝直接调用商户指定的 call_back_url，并携带参数；异步是指支付宝在支付完成后发送通知到商户指定的 notify_url，以下为具体内容。

2.4.1 call_back_url

用户在支付宝收银台完成支付后，会以 GET 方式跳转到 call_back_url（用户直接点击或自动跳转），同时会携带交易参数。商户在收到这一参数后，要先进行验签。样例如下：

样例

```
http://10.14.42.49:8080/paychannel/servlet/CallBack?out_trade_no=1320742949342&request_token=requestToken&result=success&trade_no=2011110823389231&sign=49a330fee069465c64e561a25bf31c78
```

商户可根据“result”参数判断交易状态。具体参数的含义请查询参数表

2.4.2 notify_url

在交易完成后，支付宝会通过 Post 请求该地址，将交易状态信息发送给商户服务器。

商户通过支付宝的通知状态（trade_status）判断交易是否成功，具体如下：

商户地址：提供一个 http 的 URL(例:http://www.partnertest.com/servlet/NotifyReceiver)，支付宝将以 POST 方式调用该地址。

通知触发条件：交易状态发生改变，如交易从“创建”到“成功”或“关闭”。

商户返回信息：商户服务器收到通知后需返回纯字符串“success”，不能包含其他任何 HTML 标签等字符。

通知重发：若支付宝没有收到商户返回的“success”，将对同一笔订单的通知进行周期性重发 (间隔时间为：2 分钟,10 分钟,10 分钟,1 小时,2 小时,6 小时,15 小时共 7 次)。

交易判断条件：收到 trade_status=TRADE_FINISHED（如果签有高级即时到帐协议则 trade_status=TRADE_SUCCESS）的请求后才可判定交易成功（其它 trade_status 状态请求可以不作处理）

以下为支付宝通知的样例：

1） RSA 方式签名时

当商户使用 RSA 签名方式时，商户实际收到支付宝通知请求如下：

样例：

http://www.partnertest.com/servlet/NotifyReceiver?service=alipay.wap.trade.create.direct&sign=Rw/y4ROnNicXhaj287Fiw5pvP6viSyg53H3iNiJ61D3YVi7zGniG2680pZv6rakMCeXX++q9XRLw8Rj6I1//qHrwMAHS1hViNW6hQYsh2TqemuL/xjXRCY3vjm1HCoZOUa5zF2jU09yG23MsMIUx2FAWCL/rgbcQcOjLe5FugTc=&sec_id=0001&v=1.0&notify_data=g3ivqicRwI9rI5jgmSHSU2osBXV1jcxohapSAPjx4f6qiqsoAzstaRWuPuutE0gxQwzMOtwL3npZqWO3Z89J4w4dXIY/fvOLoTNn8FjExAf7OozoptUS6suBhdMyo/YJyS3lVALfCeT3s27pYWihHgQgna6cTfgi67H2MbX40xtexIpUnjgxBkmOLai8DPOUI58y4UrVwoXQgdcwnXsfn2OthhUFiFPfpINgEphUAq1nC/EPymP6ciHdTCWRI6l1BgWuCzdFy0MxJLliPSnuLyZTou7f+Z5Mw24FgOacaISB+1/G+c4XIJVKJwshCDw9Emz+NAWsPvq34FEEQXVAeQRDOphJx8bDqLK75CGZX+6fx88m5ztq4ykuRUcrmoxZLJ+PiABvYFzi5Yx2uBMP/PmknRmj1HUKEhuVWsXR0t6EWpJFXlyQA4uxbShzncWDigndD7wbfNtkNLg5xMSFFIKay+4YzJK68H9deW4xqk4JYTKsv8eom9Eg9MrJZiIrFkFpVYPuaw0y/n61UEFYdzEQZz+garCmMYehEAQCGibYUQXBlf1iwTOZdqJIxdgCpSX21MIa9N9jicmFu8OXWZJkdN+UrSyvIcpzRori+U6522ovMz5Z8EzVTfcUENu+dWJRnhFlo6pvm0a3Fq2wBEyUV1/YYS3LaZiPj+wig5BCyJ92QXZnEUEtn87oX5kuzSRuLcVVi8OJIgyQWaWT9N0YFyH5AfV+VDNxu4UYy6KkGtcaVjSvbzDuzThMXs2HDwX3qujq25A/hzJKlgR9EjcumJeF/TM6eS7JS+FKXE1kUXnMnGbokaNemZn2yKlPC1VO4LU77G5v1nUs6MfYFq9HC4FYiQ6Y+hL8RgAMorty/RYT3cZ8SQCTO0bQ+qJuOnx79YEEmCUQc3iJBp0zFKYXIU6viqJYghEs6F3LiK8TvJRo8+ST5hKtnuU5b8R6f9yD8Uek1BruWvlYaA7I3Cc90CDhTyOghL2oCMOoKlxqgXdh3MGm128FOVyCjDLRw04b+kK83JGFMcjyVuhfhoVeETQicUCtFQ9ItlH3uFkB5su+r3399WGSXyGflrTbFhMq7mRztWotL2ATvf/enMBcGSCSCb47OzGxXhMDGZZu4Sq4pdF9fsZVBHgWsB/KS8bwxyvM068NoqnRmI72zgL7WFWumlm88j3K6KPxbB6soDSXRv6drbSv2t93llE5q4SP6GLztAw7UPWGTJLXOFyhyaszvhyZWxsX+C5PbXoCta1/cxt4Sp4WXDjaHn6qHI/Vea28xx8fYV/xK5WTmvFwb0k9eRGCgB6/nzmGV1+lPJuK3pKy3L5LbUP0zJFh5gdPG7DecH+F0uBUC0QNMQ==

其中 notify_data 参数值为加密内容，商户需用商户 RSA 私钥先进行解密后再验签，验签具体请见这里 注意：支付宝系统通知待签名数据构造规则比较特殊，为固定顺序。

2） MD5 方式签名时

当商户使用 MD5 签名方式时，商户实际收到支付宝通知请求如下

样例：

http://www.partnertest.com/servlet/NotifyReceiver?service=alipay.wap.trade.create.direct&sign=Rw/y4ROnNicXhaj287Fiw5pvP6viSyg53H3iNiJ61D3YVi7zGniG2680pZv6rakMCeXX++q9XRLw8Rj6I1//qHrwMAHS1hViNW6hQYsh2TqemuL/xjXRCY3vjm1HCoZOUa5zF2jU09yG23MsMIUx2FAWCL/rgbcQcOjLe5FugTc=&v=1.0&sec_id=MD5&notify_data=<notify><payment_type>1</payment_type><subject>收银台【1283134629741】</subject><trade_no>2010083000136835</trade_no><buyer_email>dinglang@a.com</buyer_email><gmt_create>2010-08-30 10:17:24</gmt_create><notify_type>trade_status_sync</notify_type><quantity>1</quantity><out_trade_no>1283134629741</out_trade_no><notify_time>2010-08-30 10:18:15</notify_time><seller_id>2088101000137799</seller_id><trade_status>TRADE_FINISHED</trade_status><is_total_fee_adjust>N</is_total_fee_adjust><total_fee>1.00</total_fee><gmt_payment>2010-08-30 10:18:26</gmt_payment><seller_email>chenf003@yahoo.cn</seller_email><gmt_close>2010-08-30 10:18:26</gmt_close><price>1.00</price><buyer_id>2088102001172352</buyer_id><notify_id>509ad84678759176212c247c46bec05303</notify_id><use_coupon>N</use_coupon></notify>

其中 notify_data 参数值为明文内容，无需解密。

通知中其他参数意义详见参数列表

第三章签名详解

3.1RSA 和 openssl 简介

3.1.1 什么是 RSA

RSA 是一种非对称的签名算法，即签名密钥（私钥）与验签密钥（公钥）是不一样的，私钥用于签名，公钥用于验签。

在与支付宝交易中，会有 2 对公私钥，即商户公私钥，支付宝公私钥。

3.1.2 为什么要用 RSA

使用这种算法可以起到防止数据被篡改的功能，保证支付订单和支付结果不可抵赖(商户私钥只有商户知道)。

3.1.3 什么是 OpenSSL

一句话概括：OpenSSL 是基于众多的密码算法、公钥基础设施标准以及 SSL 协议安全开发包。

3.1.4 为什么要用 OpenSSL

通过 OpenSSL 生成的签名和内置的算法可以做到跨平台，这样在不同的开发语言中均可以签名和验签。

3.2 RSA 密钥详解 *

3.2.1 找到生成 RSA 密钥工具

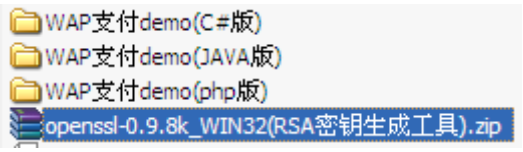
步骤 1

下载开发指南和集成资料，如下图，您能看到此文档说明开发指南和集成包已经下载了。

图：3-1 下载开发指南和集成资料

步骤 2

解压下载的压缩包(WS_WAP_PAYWAP)，找到并解压 openssl-0.9.8k_WIN32(RSA 密钥生成工具).zip 工具包



图：3-2 openssl 工具包

3.2.2 生成商户密钥并获取支付宝公钥

(1) 生成原始 RSA 商户私钥文件

假设解压后的目录为 c:\alipay，命令行进入目录 C:\alipay\bin，执行“*openssl genrsa -out rsa_private_key.pem 1024*”，在 C:\alipay\bin 下会生成文件 rsa_private_key.pem，其内容为原始的商户私钥（请妥善保存该文件），以下为命令正确执行截图：

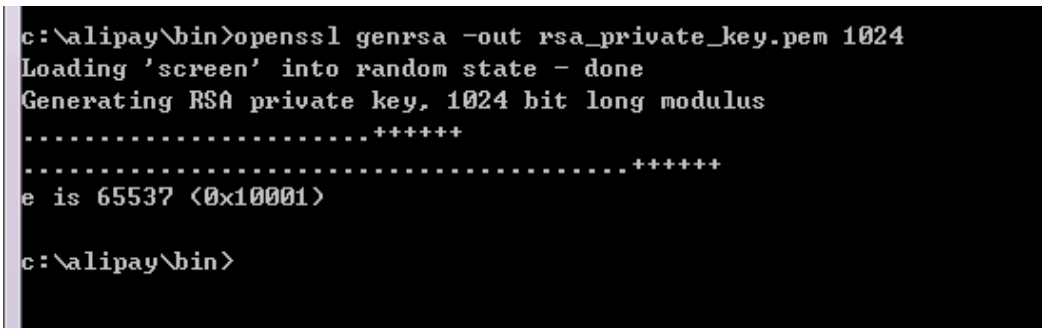


图 3-3 生成原始 RSA 商户私钥文件

(2) 生成 RSA 商户公钥

命令行执行“ *openssl rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem*”，在 C:\alipay\bin 文件夹下生成文件 rsa_public_key.pem。接着用记事本打开 rsa_public_key.pem，复制全部内容至新建的 txt 文档，删除文件头“-----BEGIN PUBLIC KEY-----”与文件尾“-----END PUBLIC KEY-----”及空格、换行，如下图。最后得到一行字符串并保存该 txt 文件为“public_key.txt”。

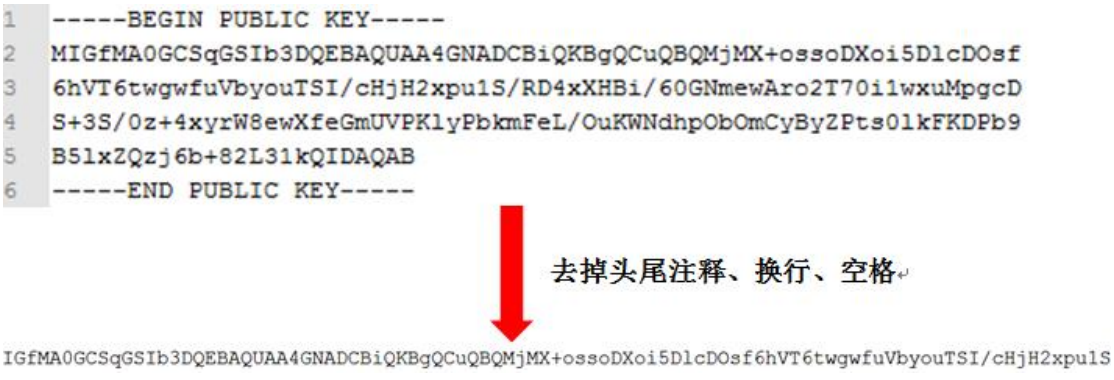


图 3-5 生成公钥

(3) 上传商户公钥至支付宝

浏览器访问 https://ms.alipay.com/index.htm 并用签约帐号登录，点击菜单栏“我的商家服务”，右侧点击“密钥管理”，见下图红色框内

图 3-6 商户公钥上传

点击“上传”，选择步骤(3)生成的“public_key.txt”并完成上传。

(4) 获取 RSA 支付宝公钥

成功上传公钥至支付宝后，页面显示如下：

图 3- 7 支付宝公钥获取

其中红色框内部分即支付宝公钥，请复制至新建 txt 文档，将文档中的空格删除并敲打回车。

3.3 RSA 签名和验签 *

3.3.1 RSA 签名

定义：

RSA 是一种非对称的签名算法，即签名密钥（私钥）与验签名密钥（公钥）是不一样的，私钥用于签名，公钥用于验签名。使用这种算法签名在起到防数据篡改功能的同时，还可以起到防抵赖的作用，因为私钥只有签名者知道。

核心代码是调用 alipay_function.php 文件中的 sign 方法，创建签名用的是商户的私钥。

```
/**RSA 签名
 * $data 签名数据(需要先排序，然后拼接)
 *  签名用商户私钥，必须是没有经过 pkcs8 转换的私钥
 *  最后的签名，需要用 base64 编码
 * return Sign 签名
 */
function sign($data) {
    //读取私钥文件
    $priKey = file_get_contents('key/rsa_private_key.pem');

    //转换为 openssl 密钥，必须是没有经过 pkcs8 转换的私钥
    $res = openssl_get_privatekey($priKey);

    //调用 openssl 内置签名方法，生成签名$sign
openssl_sign($data, $sign, $res);

    //释放资源
openssl_free_key($res);

    //base64 编码
    $sign = base64_encode($sign);
    return $sign;
}
```

3.3.2RSA 验签

核心代码是调用 alipay_function.php 文件中的 verify 方法，验签方法中用的是支付宝公钥。

```
/**RSA 验签
 * $data 待签名数据(需要先排序，然后拼接)
 * $sign 需要验签的签名,需要 base64_decode 解码
 * 验签用支付宝公钥
 * return 验签是否通过 bool 值
 */
function verify($data, $sign)  {
    //读取支付宝公钥文件
    $pubKey = file_get_contents('key/alipay_public_key.pem');

    //转换为 openssl 格式密钥
    $res = openssl_get_publickey($pubKey);

    //调用 openssl 内置方法验签，返回 bool 值
    $result = (bool)openssl_verify($data, base64_decode($sign), $res);

    //释放资源
openssl_free_key($res);

    //返回资源是否成功
    return $result;
}
```

3.3.3 RSA 解密

核心代码是调用 alipay_function.php 文件中 decrypt 方法

```
/**解密
 * $content为需要解密的内容
 * 解密用商户私钥
 * 解密前，需要用base64将内容还原成二进制
 * 将需要解密的内容，按128位拆开解密
 * return 解密后内容，明文
 */
function decrypt($content) {

    //读取商户私钥
    $priKey = file_get_contents('key/rsa_private_key.pem');

    //转换为openssl密钥，必须是没有经过pkcs8转换的私钥
    $res = openssl_get_privatekey($priKey);
```

```
    //密文经过base64解码
    $content = base64_decode($content);

    //声明明文字符串变量
    $result  = '';

    //循环按照128位解密
    for($i = 0; $i <strlen($content)/128; $i++  ) {
        $data = substr($content, $i * 128, 128);

        //拆分开长度为128的字符串片段通过私钥进行解密，返回$decrypt解析后的明文
openssl_private_decrypt($data, $decrypt, $res);

        //明文片段拼接
        $result .= $decrypt;
    }

    //释放资源
openssl_free_key($res);

    //返回明文
    return $result;
}
```

3.4 MD5

3.4.1 MD5 简介

定义：

MD5 是一种摘要生成算法，本来是不能用于签名的。但是，通过在待签名数据之后加上一串私密内容（指令发送、接收双方事先规定好的，这里我们称其为签名密钥），就可以用于签名了。使用这种算法签名只能起到防数据篡改的功能，不能起到签名防抵赖的功能，因为双方都知道签名密钥

3.4.2 MD5 Key

当商户使用 MD5 加密方式生成签名之前，需要将待签名参数加上 MD5 Key 参数。获取 Key：登录 https://ms.alipay.com 我的商家服务->密钥管理，然后复制出来 MD5 密钥字

符串，如下图

图：3-8 复制 MD5 密钥

3.4.3 MD5 签名和验签

签名：调用 alipay_function.php 的 sign_MD5 方法，代码如下：

```
/**MD5 签名方法
 * $prestr 需要签名的字符串
 * $sign_type 签名类型，也就是 sec_id
 * return 签名结果
 */
function sign_MD5($prestr,$sign_type) {
    $sign='';
    if($sign_type == 'MD5') {
        $sign = md5($prestr);
    }elseif($sign_type =='DSA') {
        //DSA 签名方法待后续开发
        die("DSA 签名方法待后续开发，请先使用 MD5 签名方式");
    }else {
        die("支付宝暂不支持".$sign_type."类型的签名方式");
    }
    return $sign;
}
```

验签：获取支付宝返回的数据（除签名）并调用上面的 sign_MD5 方法，并对比返回的 sign 签名，如果相同代表验签通过，否则验签没有通过，可能表单已经被篡改。

附录 A 错误代码列表

错误代码	说明
0000	系统异常
0001	缺少必要的参数，检查非空参数是否已经传递
0002	签名错误，检查签名的参数是否符合支付宝签名规范
0003	服务接口错误，检查 service 是否传递正确
0004	req_data 格式不正确
0005	合作伙伴没有开通接口访问权限，合同是否有效
0006	sec_id 不正确，支持 0001，MD5
0007	缺少了非空的业务参数
ILLEGAL_SIGN	签名错误，检查签名的数据是否符合支付宝签名规范
ILLEGAL_SERVICE	接口不存在，检查 service 是否传递正确
ILLEGAL_PARTNER	无效商户，检查传入的 PARTNER 值是否正确
ILLEGAL_PARTNER_EXTERFACE	商户接口不存在，该商户没有开通该接口
HAS_NO_PRIVILEGE	无权访问该接口
SYSTEM_ERROR	系统错误

附录 B 手机网站支付接口参数表

参数名	中文描述	类型(精度)	说明	商户必传	参数值样例
service	接口名称	String	注意：交易创建、授权并执行两次请求的值不同。	Y	alipay.wap.trade.create.direct/alipay.wap.auth.authAndExecute
partner	合作伙伴 id	String(16)	合作伙伴在支付宝的用户 ID，与支付宝签约后自动生成	Y	2088002007015955
sec_id	签名算法	String(4)	签名算法。目前只支持 MD5 和 RSA(用 0001 表示)	Y	0001 或 MD5
req_id	请求号	String(32)	请求号用于关联请求与响应，并且防止请求重播。支付宝 wap 限制来自一个 partner 的	Y	1e925b9b4b115961660130f9281e3898

			请求号必须唯一。		
sign	签名	String	签名，对 request/response 中参数签名后的值	Y	72020eb70e0fdcfbf404edcbb83bfd81
format	请求参数格式	String	参数值必须和样例保持一致	Y	xml
v	接口版本号	String	参数值必须和样例保持一致	Y	2.0
req_data	请求业务参数	String	参数值内容为 xml 格式,包含内层标签参数	Y	<?xml version="1.0" encoding="UTF-8"?><direct_trade_create_req> <subject>彩票</subject><out_trade_no>20080801-1</out_trade_no><total_fee>50</total_fee><seller_account_name>tbbusi003@126.com</seller_account_name><out_user>xxxxx</out_user><notify_url>http://www.yoursite.com/notifyurl.htm</notify_url></direct_trade_create_req>
direct_trade_create_req	固定标签	String	req_data 参数值 xml 内容中必须包含的固定标签。	Y	<subject>彩票</subject><out_trade_no>20080801-1</out_trade_no><total_fee>50</total_fee><seller_account_name>tbbusi003@126.com</seller_account_name><out_user>xxxxx</out_user><notify_url>http://www.yoursite.com/notifyurl.htm</notify_url>
subject	商品名称	String(256)	订单商品名称	Y	彩票
out_trade_no	外部交易号	String(64)	合作伙伴系统的交易号，传递给支付宝系统做外部交易号（不能重复）	Y	2008080101
total_fee	订单价格	String(15)	用户购买的商品或服务的价格（必须是金额的格式,单位：元）	Y	1.01
pay_expire	交易自动关闭时间	Int	买家如未能在该设定值范围内支付成功，交易将被关闭。单位：“分钟”，值区间	N	10

			0<pay_expire，默认值 21600（15 天）。最终关闭时间点误差 1-2 分钟。		
seller_account_name	卖家帐号	String(100)	交易卖方的支付宝帐号，交易成功后该笔交易的资金会转入这个支付宝帐号中	Y	tbbusi003@126.com
out_user	商户系统用户唯一标识	String(32)	买家在商户系统的唯一标识，当该 out_user 支付成功一次后再来支付时，30 元内无需密码。	N	21321211111
notify_url	商户接受通知的 url	String(200)	商户接受通知的 url	Y	http://www.yoursite.com/notifyurl.htm
merchant_url	返回商户链接	String	用户在支付宝页面可返回商户的链接	N	http://www.yoursite.com/partnerurl.htm
call_back_url	支付成功跳转链接	String(200)	由商户提供，只有当交易支付成功之后，才会跳转到该 url。	Y	http://www.yoursite.com/callbackurl.htm
cashierCode	支付前置银行代码	String	调用支付前置接口，由支付宝返回给商户所支持的银行代码	N	CREDITCARD_ICBC
request_token	token	String(40)	前面调用交易创建接口成功返回后获得的**（注当此参数为页面返回时，为固定值）**		20081113f9d49c20e8e5c8e40b6107ec42259e41
trade_no	交易号	String(64)	交易号，该笔交易在支付宝系统的交易号		2009092904171521
notify_data	通知业务参数	String	通知的业务参数，包含交易号、外部交易号、交易状态等信息。		见例子
payment_type	支付方式	String	用户的支付方式(商户可不关心该参数)		1
buyer_email	买家账号	String(100)	买家的支付宝账号		chenf002@yahoo.cn
gmt_create	创建时间	String	交易创建时间		2009-09-29 19:59:24
notify_type	通知类型	String	该通知的类型，暂时只有交易状态同步(商户可不关心该参数)		trade_status_sync
quantity	数量	String	购买商品数量		1
notify_time	通知时间	String	发送通知的时间		2009-09-29 19:59:25
seller_id	卖家 id	String	卖家的支付宝账号 id		2088102001058148

trade_status	交易状态	String	交易的状态。TRADE_FINISHED（支付成功），WAIT_BUYER_PAY(等待买家付款)		TRADE_FINISHED/ WAIT_BUYER_PAY
is_total_fee_adjust	总价是否被修改	String	交易价格是否被修改，Y 或 N(本接口创建的交易不会被修改)		N
total_fee	交易总价	String	即订单金额。单位：元		2.21
gmt_payment	付款时间	String	交易的付款时间，如果交易未付款，没有该属性		2009-09-29 19:59:25
seller_email	卖家账号	String(100)	卖家的支付宝账号		youngbeckham@gmail.com
gmt_close	交易结束时间	String	交易结束的时间		2009-09-29 19:59:25
price	单个商品价格	String	目前和 total_fee 值相同。单位：元		2.21
buyer_id	买家 id	String	买家的支付宝账号 id		2088101000137393
notify_id	通知 id	String	唯一识别通知内容，重发相同内容的通知 notify_id 值不变。		2311b764be6fba98f593ba98f7eb7470
use_coupon	是否使用红包	String	交易时是否使用红包，Y 或 N		N
_input_charset	参数编码字符集	String	见签名机制		UTF-8