

**Rules on Access and Use**

**Minimal Checkout SDK Payment Interface  
Appendix**

**Version No.: 1.1**

Alipay (China) Network Technology Co., Ltd. All Rights Reserved

Table of Contents

[1 File Description 错误!未定义书签。](#_Toc387137253)

[1.1 File Description **错误!未定义书签。**](#_Toc387137254)

[1.2 Service Terms **错误!未定义书签。**](#_Toc387137255)

[2 Attribution of Liability 错误!未定义书签。](#_Toc387137256)

[3 Rules on Technology Access 错误!未定义书签。](#_Toc387137257)

[Access Procedures 错误!未定义书签。](#_Toc387137258)

[3.1 General Access Procedures **错误!未定义书签。**](#_Toc387137259)

[3.2 Notification Rules **错误!未定义书签。**](#_Toc387137260)

[3.2.1 Nonrefundable Movable Shortcut Payment **错误!未定义书签。**](#_Toc387137261)

[3.2.2 Refundable Movable Shortcut Payment **错误!未定义书签。**](#_Toc387137262)

[4 Detailed Explaination on Intergration Procedures 错误!未定义书签。](#_Toc387137263)

[4.1 Previous Preparation for Access **错误!未定义书签。**](#_Toc387137264)

[4.2 SDK Integration Procedures **错误!未定义书签。**](#_Toc387137265)

[4.2.1 iOS **错误!未定义书签。**](#_Toc387137266)

[4.2.2 Android **错误!未定义书签。**](#_Toc387137267)

[5 Rules on Testing Procedures 错误!未定义书签。](#_Toc387137268)

[6 Addendum 错误!未定义书签。](#_Toc387137269)

[6.1 Generation and Application of RSA Key **错误!未定义书签。**](#_Toc387137270)

[6.1.1 Generation of Merchant Key **错误!未定义书签。**](#_Toc387137271)

[6.1.2 Application Logic of RSA Key **错误!未定义书签。**](#_Toc387137272)

[6.2 Business Data Transmission **错误!未定义书签。**](#_Toc387137273)

# File Description

## File Description

As an appendix to *Minimal Checkout SDK Payment Interface*, this file explains some points need to pay attention in technology access and use in details so as to help merchants to avoid risks.

In case of any questions after reading, please contact relevant Alipay technology support.

## Service Terms

Table 1-1 Service Terms

|  |  |
| --- | --- |
| Term | Interpretation |
| Request | A process of transmitting data required in the form of character string by Android/iOS client to receipient party. |
| Return | Alipay returns the processing result data in the form of character string to Android/iOS client directly. |
| Notify | Asynchronous notification from server. Alipay server takes the initiative to notify partner website and feeds back the processing result to partner website after the data received is processed by Alipay. |
| Sensitive words | The words with sensitive political orientation, violence tendancy, blue or uncivilized meaning. |

# Attribution of Liability

All rules refered to in this file are precautionary measures made according to some main risks arising in access and use of Alipay Interface, please pay attention. Please strictly comply with the interface information, such as interface technical files (Movable Shortcut Payment Application Integration Access Packet Payment Interface.pdf), code examples, this file (Rules on Access and Use of Movable Shortcut Payment Application Integration Access Packet Payment Interface) and so on, provided by Alipay when accessing and using Alipay Interface, otherwise, any risks, capital loss or other estensive circumstances thus incurred shall be borne by the merchants.

# Rules on Technology Access

Table 3-1 Rules on Technology Access

| Type | Detailed Rules | Reason |
| --- | --- | --- |
| Account No. | Partner ID and security check code key of the configuration must match subscription information | To prevent the failure to use the interface normally or capital loss. |
| The privacy of partner ID and security check code key must be protected. | To prevent the account information from being stolen, resulting in capital loss or malicious use by others. |
| Testing account number must be replaced with contractual account number upon completing the test. | 3% of commission fees will be deducted when using testing account number. |
| Security | Merchant must access Alipay Interface by means of DNS resolution, Neither DNS cache must be set, nor Alipay IP must be signed up. If Alipay IP must be signed up for the security of merchant, records must be filed with Alipay technical support personnel. | Once the Alipay IP adreess changes, merchant will be unable to request or access Alipay, which will make the merchant service unavailable directly. |
| Signature | The character string to be signed shall be spliced according to the rules of “parameter name 1 = parameter value 1 & parameter 2 = parameter 2&…&parameter name N = parameter value N”. | To prevent the failure to use the interface normally |
| When signing on request parameter, the request parameter must come from request parameter list, and the parameter “sign” in the list must be removed. | To prevent the failure to use the interface normally |
| When signing on request parameter, for the nullable parameter in request parameter list, if being used, its parameter value must not be null or null value. | To avoid reporting abnormal error, please refer to Error Code List for the error code. |
| Parameter Configuration | In Request Parameter List, unnullable parameter must be configured. | To prevent the failure to use the interface normally |
| In Request Parameter List, at least one nullable parameter subject to multiple choices must be configured. | To prevent the failure to use the interface normally |
| Configuration must be subject to the format requirements on all kinds of parameters in Request Parameter List. | To prevent the failure to use the interface normally |
| Request parameter (\_input\_charset) (encoding format) must be confirgured, which means this parameter cannot be nullable, and shall take part in signature algorithm, and the value of this parameter can only be set as utf-8, which means this product does not support GBK encoding format. | To avoid reporting abnormal error, for example, the signature is incorrect. |
| Seller is the Alipay account number when making collections, the same one with the Alipay account number which needs to correspond to partner, that is to say Alipay account number for making collections must be the Alipay account number for signing contract. | To avoid the possibility of capital loss of Alipay account number for signing contract. |
| Signature mode supports RSA only. | To avoid unsuccessful signature |
| pkcs8 encoding | Movable shortcut payment requires private key of merchant to do pkcs8 encoding so as to support higher mobile system version, but php is free of such requirement. | Php does not support pkcs8. |
| Interface structure | Server-side: generate and submit parameters as well as deal with asynchronous notification return of Alipay. | SDK is mde up of server-side and client-side. In consideration of the security of interaction information, usually, the parameter is stored on the server-side, which can be obtained from server-side if client-side needs. |
| Client-side: build up form parameters, which shall be submitted to Alipay. |
| When submitting payment parameters, order information (orderInfo) needs to be packaged, thereinto, the parameter shall be presented in the form of key=”value”, and “&” shall be used for dividing parameters, obtainning Alipay payment object to call payment service. | To avoid reporting error when requesting Alipay, the error code is incorrect signature. |
| Data transmission | https protocol must be used. | To prevent the failure to use the interface normally |
| Notification return verification | In asynchronous notification from the server-side in SDK payment interface, when signing on notified parameters, the parameters must come from the parameters returned upon notification of Alipay, and the parameter “sign” in the list must be removed. Firstly, sort the above said parameters in the form of “parameter name = parameter value” from in alphabetical order from a to z, then splice them according to the rules of “parameter name 1 = parameter value 1 & parameter 2 = parameter 2&…&parameter name N = parameter value N”, make a comparision between the signature results from the above and the parameter “sign” value obatained. | Verify the signature returned. |
| Disposal of returned data | Alipay takes the initiative to send notification. Upon receiving such notification, merchant must return “success” character string to Alipay instead of other characters. | If the information returned by merchant to Alipay is not “success”, Alipay may repeatedly send notification 7 times at most. |
| Notification path set must be accessed through Internet, and can be access smoothly. | To avoid the failure to receive the notification sent by Alipy. |
| Returned data must be disposed. | To help merchant to know the use of interface so that merchant can conduct subsequent business operation. |
| In page files asynchronously notified by server-side, only when all businesses of merchant are operated and completed, print can be made. | To avoid abnormal asynchronous notification. For example, the notification cannot be received or notify Alipay system that the business processing is completed when it is not. |
| It is suggested to record every business operation in the log operation database in merchant websit in the form of log, set up a mechanism to judge repeated notification. | Check or follow up business processing status if neccessary |
| Rules on writing interface code automously | If the interface is integrated without using the code examples provided by Alipay, the interface code applicable to merchant website program must be encoded according to the signature mechanism set out in technical files and notification return data disposal chapter, rules on technology access, rules on interface use, rules on testing procedures specified in this file. | To prevent the failure to use the interface normally |

# Access Procedures

## General Access Procedures

Choose Alipay to make payment



Application procedures for independent payment or safe payment with Alipay client-side

Procedures for calling SDK

Application procedures for embedded SDK

Application procedures for WAP payment

Return to original application

Return to original application

Figure 3-e General SDK Access Procedures

## Notification Rules

### Nonrefundable Movable Shortcut Payment

The asynchronous notification of movable shortcut payment involves two notified trading status (trade\_status).

* The first status value is WAIT\_BUYER\_PAY, which means waiting for payment.Merchant may conduct relevant operations based on its business logic demand, and return SUCCESS character string to Alipay after completing business logic;
* The second status value is TRADE\_FINISHED, which means the transaction is done, that is to say the payment for this order is made on Alipay side successfully.Merchant may conduct relevant business logic operations based on this status, and return SUCCESS character string to Alipay at last.

### Refundable Movable Shortcut Payment

The mechanism of refundable asynchronous notification is consistent with that of nonrefundable asynchronous notification, the first status of them (WAIT\_BUYER\_PAY) is the same, the second status is TRADE\_SUCCESS, under which merchant may conduct relevant business logic operations and return SUCCESS. The third status is TRADE\_FINISHED, which means the order comes to an end, and the payment cannot be refunded.

Judge whether the commission fees will be refunded based on notification:

* Refund commission fees: After refund operation for a single transaction is finished, Alipay asynchronous notification sends the status of TRADE\_CLOSE (transaction is closed). This transaction status shall be opened by Alipay backstage configuration alone, whosedefault status is closed.
* Not refund commission fees: After refund operation for a single transaction is finished, Alipay asynchronous notification sends the status of TRADE\_SUCCESS, and send the status of TRADE\_FINISHED within three months after the payment for order is completed.

# Detailed Explaination on Intergration Procedures

## Previous Preparations for Access

Previous preparations for access include merchant contract and key confirguration, except for those merchants who have aleady finished the above preparations.

## SDK Integration Procedures

### iOS

#### Add AlipaySDK.framework in AlipaySDK.embeddedframework into Link Binary With Libraries.

#### Include the header file #import <AlipaySDK/Alipay.h>。

#### Encoding order information, the parameters are as follows, please refer to Demo for details:

Order \*order = [[Order alloc] init];

order.partner = partner;

order.seller = seller;

order.tradeNO = [self generateTradeNO]; //order ID (drawn up by marchant)

order.productName = product.subject; //product title

order.productDescription = product.body; //product description

order.amount = [NSString stringWithFormat:@"%.2f",product.price]; //product price

order.notifyURL = @"http://www.xxx.com"; //callback URL

order.uid = @"3593190900";

order.externChannel = @"sina";

order.accessCode = @"xxxxxxxxxxxxxxxxxxx";

order.service = @"mobile.securitypay.pay";

order.paymentType = @"1";

order.inputCharset = @"utf-8";

order.itBPay = @"30m";

order.showUrl = @"m.alipay.com";

#### Covert order information into character string, sign on character string, and splice the same in accordance with the following form:

NSString \*orderSpec = [order description];

NSString \*signedString = [signer signString:orderSpec];

NSString \*orderString = nil;

if (signedString != nil) {

orderString = [NSString

stringWithFormat:@"%@&sign=\"%@\"&sign\_type=\"%@\"",

orderSpec, signedString, @"RSA"];

}

#### Call payment service

[[Alipay defaultService] pay:orderString

From:appScheme

CallbackBlock:^(NSString \*resultString) {

NSLog(@"result = %@",resultString);// payment result returned

}];

Description: appScheme is the scheme registered by app in info.plist.

If the embedded minimal checkout is unavailable, SDK will jump to Alipay Wallet to complete payment and jump back to calling party “app” after payment.

In- (BOOL)application:(UIApplication )application openURL:(NSURL )url sourceApplication:

(NSString \*)sourceApplication annotation:(id)annotation, call

[[Alipay defaultService]

processPayResultFromAlipayclientWithOrder:url];

SDK will make a uniform callback in the above said CallbackBlock after completing analysis on payment results.

#### Acquisition and Disposal of Notification from Alipay Server

Merchant needs to provide an interface for https protocol, which shall be included in the parameter, and transmit such interface to shortcut payment, i.e. notify\_url. Alipay server will call notify\_url by means of POST upon payment, and transmit payment result. Please refer to asynchronous notification parameter description set out in interface file for details.

### Android

#### Development Kit

Minimal SDK is integrated into the client projects of merchant’s applications in the form of plug-in unit, and it is libraryproject alipay\_lib.

#### Import Development Resources

Unzip alipay\_lib.zip, copy the alipay\_lib after unzipping into Eclipse workspace, import this project via Eclipse, and select this project as library project through Properties->Android of this project, as shown in the following figure:

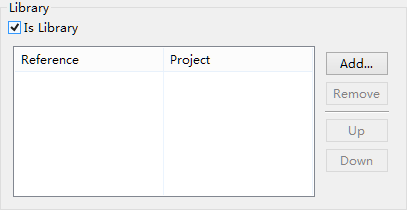


Figure 4-1 Resources Importing Project

Add in the Properties->Android of merchant application projects, as shown in the following figure:

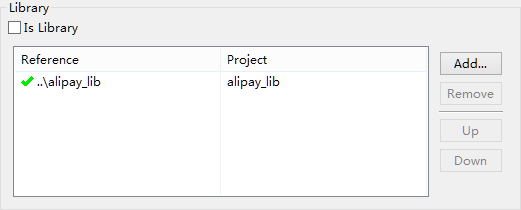


Figure 4-2 Add library

Figure 4-3 Add alipay.jar

#### Modify Manifest

Add statement into AndroidManifest.xml file of merchant application projects:

<activity android:name="com.alipay.android.mini.window.sdk.MiniLaucherActivity"

android:configChanges="orientation"

android:excludeFromRecents="true"

android:launchMode="singleTop"

android:theme="@style/MspAppTheme" />

<activity

android:name="com.alipay.android.mini.window.sdk.TransContainer"

android:configChanges="orientation"

android:excludeFromRecents="true"

android:launchMode="singleTop"

android:theme="@style/MspAppTheme" />

<activity android:name="com.alipay.android.mini.window.sdk.MiniPayActivity"

android:configChanges="orientation"

android:excludeFromRecents="true"

android:label="@string/msp\_app\_name"

android:launchMode="singleTop"

android:theme="@style/MspAppTheme"

android:windowSoftInputMode="adjustResize" >

<intent-filter android:priority="800" >

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

<activity

android:name="com.alipay.android.mini.window.sdk.MiniWebActivity"

android:configChanges="orientation|keyboardHidden|navigation"

android:excludeFromRecents="true"

android:launchMode="singleTop"

android:theme="@style/MspAppTheme" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

<service android:name="com.alipay.android.app.MspService" />

<receiver

android:name="com.ut.device.BQueryWhoHasOne"

android:exported="true"

android:permission="com.ut.permission.DEVICE\_STATE" >

<intent-filter>

<action android:name="UT.QueryWhoHasOne" />

</intent-filter>

</receiver>

<receiver

android:name="com.ut.device.BFoundIt"

android:exported="true"

android:permission="com.ut.permission.DEVICE\_STATE" >

<intent-filter>

<action android:name="UT.FoundIT" />

</intent-filter>

</receiver>

And add statement on limits of authority:

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

<uses-permission android:name="android.permission.SEND\_SMS" />

<uses-permission android:name="android.permission.READ\_SMS" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="com.ut.permission.DEVICE\_STATE" />

<uses-permission android:name="android.permission.WRITE\_SETTINGS" />

<uses-permission android:name="android.permission.READ\_SETTINGS" />

<permission android:name="com.ut.permission.DEVICE\_STATE" />

With this, SDK development resources import is finished.

#### Generation of Order Data

In minimizing SDK, order information (info) needs to be submitted, thereinto, the parameters shall be presented in the form of key=”value”, “&” shall be used for dividing parameters, all parameters are indispensible.

#### Interface Call

Obtain PayTask payment object, call payment service.

Code Example:

obtain order, build up character string

final String orderInfo = getOrderInfo( );

PayTask payTask = new PayTask(activity, new OnPayListener() {

@Override

public void onPaySuccess(Context context, String resultStatus,

String memo, String result) {

}

@Override

public void onPayFailed(Context context, String resultStatus,

String memo, String result) {

}

});

payTask.execute(orderInfo);

#### Acquisition and Disposal of Payment Result

Upon payment by calling the method of pay, the payment results will be acquired through two approaches:

* Synchronous Return

The payment results returned by the merchant application client through calling back OnPayListener will be disposed directly in application.

* Asynchronous Notification

Merchant is required to provide an interface for https protocol, which shall be included in the parameters, and transmit such interface to shortcut payment, i.e. notify\_url. Alipay server will call notify\_url by means of POST upon payment, and transmit payment results in the form of xml data.

# Rules on Testing Procedures

Table 5-1 Rules on Testing Procedures

| Step | Debugging Details | Remarks |
| --- | --- | --- |
| **Step I**: Debug this interface solely on the native machine | * Normally obtain authorization token * Normally call forth the client payment service | Only configure the interface well and do not put it in the official app project of merchant. |
| **Step II**: Debug this interface solely on the server | * Normally obtain authorization token * Normally call forth the client payment service * Synchronous return of alixpay * Return asynschronous notification from server | If there is no trouble for the interface debugging on the native machine, debug this interface on the server. |
| **Step III**: Integrate the interface into app project | N/A | Link and integrate the debugged interface with the business procedures abount the app project of merchant. |
| **Step IV**: Debug the app project after being integrated on the native machine | * Operational procedures for the whole business * Normally call forth the client payment service * Synchronous return of alixpay * Return asynschronous notification from server * Subsequent execution of business logic | Interface after the debugging on the native machine links with the app project and the distinction of client-side and server-side is completed. |

 Notice:

When testing with an account number, the authorization page may appear upon the first access to payment procedure, but it will not reappear after the setup of password-exempted limit and SMS verification.

# Addendum

## Generation and Application of RSA Key

### Generation of Merchant Key

#### Open Openssl Key Generation Software

Oepn the folder of bin under the folder of openssl and execute the file of openssl.exe, as shown in the following figure:

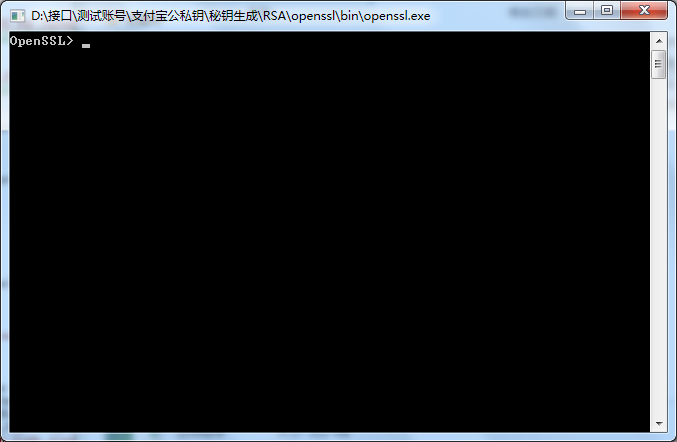


Figure 6-1 Execution of File of openssl.exe

#### Generation of RSA Private Key

Input the command of “*genrsa -out rsa\_private\_key.pem 1024*”, and hit enter, after which a file of rsa\_private\_key.pem will be newly added in the current bin file catalog, which is the original Private Key of Merchant (please keep this file properly, which is required to be used in the PHP Development Language), the following is the screenshot concerning the correct execution of commands:

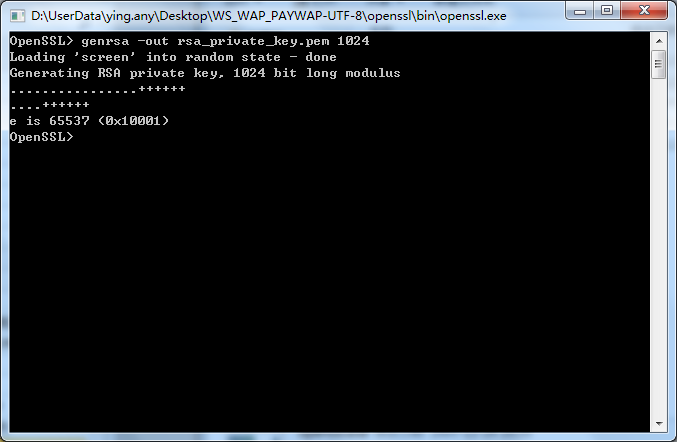


Figure 6-2 Generation of RSA Private Key

#### Generation of RSA Public Key

Input the command of “*rsa -in rsa\_private\_key.pem -pubout -out rsa\_public\_key.pem*”, and hit enter, after which a file of rsa\_public\_key.pem may be newly added in the current bin file catalog, which is the original Public Key of Merchant (please keep this file properly, which is required to be used in the PHP Development Language), the following is the screenshot concerning the correct execution of commands:

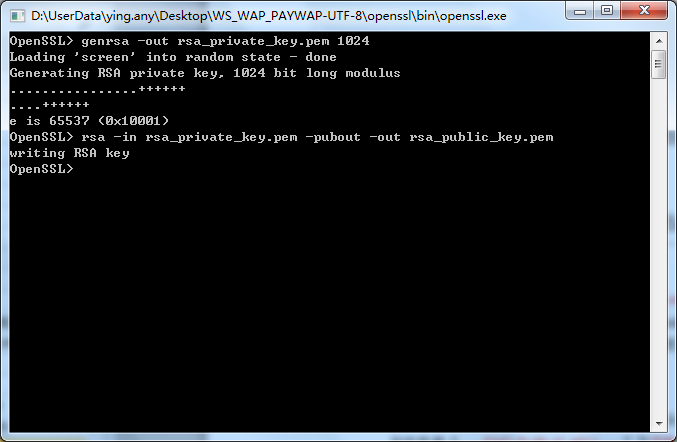


Figure 6-3 Generation of RSA Public Key

#### Generation of Private Key of PKCS8 Encoding

Input the command of “*pkcs8 -topk8 -inform PEM -in rsa\_private\_key.pem -outform PEM -nocrypt*” and hit enter, the generation results can be indicated directly in current interface:

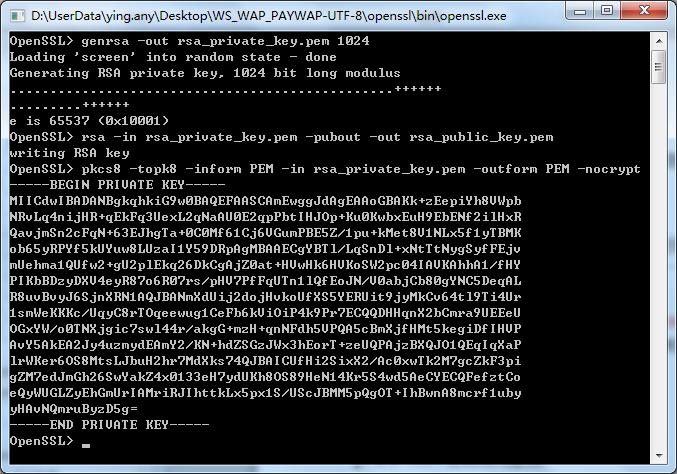
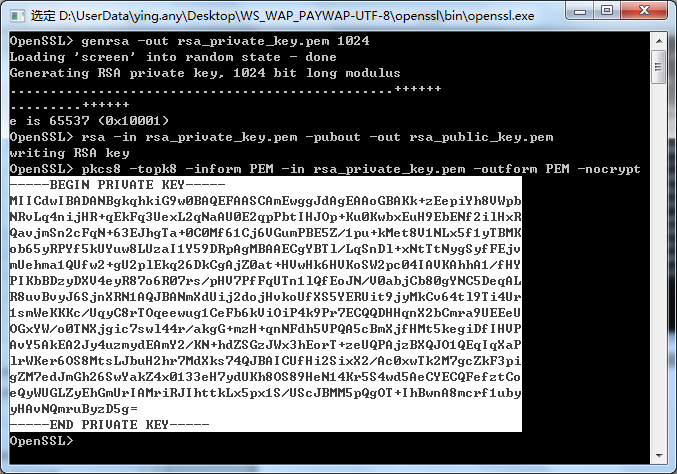


Figure 6-4 Generation of Private Key of PKCS8 Encoding

Right click of the top margin of openssl window to choose the item of “edit→tab” and then select the texts need to be copied:



Firgure 6-5 Select the Characters Need to Be Copied

Now, continue to right click the top margin of openssl window to choose “eidt→copy” and paste the copied content into a new notepad, which can be named at random as long as you know this is the Private Key in PKCS8 format (Please keep this file properly).

### Application Logic of RSA Key

Application Logic of RSA Key：

The key that may be used in reality is the Private Key of Merchant and the Public Key of Alipay while the Alipay Interface with the signature mode of RSA is used by merchant. The uploading of Public Key by merchant to Alipay and granting of Public Key by Alipay to merchant refer to the interchange operation of Public Key. As a result, Alipay side will conduct siganature authentication in accordance with the Public Key uploaded by merchant while merchant signs with its own Private Key. When merchant conduct siganature authentication with the Public Key of Alipay, in a similar way, it is also on account of the signature made by Alipay with the Private Key of Aliapy.

#### Application Method of PHP Development Language

Two files (Private Key of Mechant and Public Key of Alipay) with suffix of .pem must be stored in the Folder of key.

* Private Key of Merchant
* A line of characters must be guaranteed, i.e.: no carriage return, line feed or spacing, ect;
* pkcs8 encoding for the (original) Private Key just generated is not required
* There is no need to delete “-----BEGIN PUBLIC KEY-----” and “-----END PUBLIC KEY-----”；
* In short, just maintain the content of newly generated Private Key.
* Public Key of Alipay

RSA Public Key of Alipay refers to:

-----BEGIN PUBLIC KEY-----

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCnxj/9qwVfgoUh/y2W89L6BkRA

FljhNhgPdyPuBV64bfQNN1PjbCzkIM6qRdKBoLPXmKKMiFYnkd6rAoprih3/PrQE

B/VsW8OoM8fxn67UDYuyBTqA23MML9q1+ilIZwBC2AQ2UBVOrFXfFl75p6/B5Ksi

NG9zpgmLCUYuLkxpLQIDAQAB

-----END PUBLIC KEY-----

* + - * 1. Copy the Public Key of Alipay to the newly-built notepad and name such notepad as “alipay\_public\_key.txt”;
        2. Delete the carriage return, line feed and spacing in this character string so as to leave a line of text;
        3. Add two pieces of texts of “-----BEGIN PUBLIC KEY-----” and “-----END PUBLIC KEY-----” separately in the head and tail of this Public Key Character String of Alipay;
        4. Cut this Public Key Character String of Alipay, keep 64 characters in the first, second and third line respectively and 24 characters in the fourth line, the format after being cut shall be in consistent with that of the newly generated Public Key of merchant, as shown in the following figure:

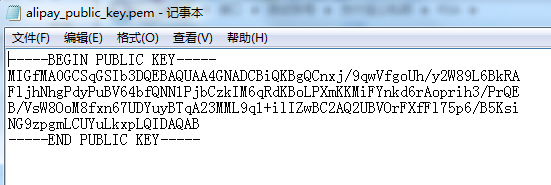


Figure 6-6 Schematic for Public Key of Alipay

* + - * 1. Save this notepad and change the suffix into .pem.

#### Application Method for JAVA and ASP.NET(C#) Dvelopment Languages

* Private Key of Merchant
* A line of characters must be guaranteed, i.e.: no carriage return, line feed or spacing, etc.；
* pkcs8 encoding for the (original) Private Key just generated is required；
* Upon completion of encoding, duplicate this Private Key and delete the carriage return, line feed, spacing therein, “-----BEGIN RSA PRIVATE KEY-----” and “-----END RSA PRIVATE KEY-----” .
* Public Key of Alipay

RSA Public Key of Alipay refers to:

-----BEGIN PUBLIC KEY-----

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCnxj/9qwVfgoUh/y2W89L6BkRA

FljhNhgPdyPuBV64bfQNN1PjbCzkIM6qRdKBoLPXmKKMiFYnkd6rAoprih3/PrQE

B/VsW8OoM8fxn67UDYuyBTqA23MML9q1+ilIZwBC2AQ2UBVOrFXfFl75p6/B5Ksi

NG9zpgmLCUYuLkxpLQIDAQAB

-----END PUBLIC KEY-----

Delete the carriage return, line feed and spacing in this character string, and a line of text must be guaranteed to be remained.

## Business Data Transmission

The business parameters supplied by Alipay refer to the data requirements needed to be transmitted by merchant to Alipay. While the business logical code is operated, merchant is only required to transmit these dynamic data to the Alipay system in the form of assignment to variable via the internal interface logic of Alipay according to its own business requirements, to be recognizable in the Alipay system.

For example, merchant needs to transmit the data concerning a certain order to Aplipay. Under such circumstance, merchant is required to get the total amount of payment (total\_fee), order No. of merchant (out\_trade\_no) and description of order (subject) and other data from its own order system in accordance with the parameter requirements of Alipay, and then assign these data one by one to corresponding variables in the form of value, combine and process such variables to a request which can be transmitted to Alipay at one time through code logic.