解密微信退款结果通知中的加密信息 req_info

作者：Ethan
原文链接：https://ld246.com/article/1560419874922
来源网站：链滴

描述

在微信返回的退款结果通知中，包含了一个加密信息字段req_info

字段名	变量名	必填	类型	示例值	描述
应用ID	appid	是	String(32)	wx8888888888888888	微信开放平台审核通过的应用APPID
退款的商户号	mch_id	是	String(32)	1900000109	微信支付分配的商户号
随机字符串	nonce_str	是	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	随机字符串，不长于32位。推荐随机数生成算法
加密信息	req_info	是	String(1024)		加密信息请用商户秘钥进行解密，详见解密方式

微信支付文档

解密方式

解密步骤如下：

（1）对加密串A做base64解码，得到加密串B

（2）对商户key做md5，得到32位小写key* (key设置路径：微信商户平台(pay.weixin.qq.com)-->账户设置-->API安全-->密钥设置

（3）用key*对加密串B做AES-256-ECB解密（PKCS7Padding）

前提工作

- **添加maven依赖**

```
<dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcprov-jdk15on</artifactId>
    <version>1.47</version>
</dependency>
```

- **替换jar包**

JAVA运行环境默认不允许256位密钥的AES加解密，解决方法就是修改策略文件

- **在官方网站下载JCE无限制权限策略文件**

JDK7版本JCE下载地址： http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html

JDK8版本JCE下载地址：http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

- 下载后解压，可以看到local_policy.jar和US_export_policy.jar以及readme.txt
- 如果安装了JRE，将两个jar文件放到%JRE_HOME%\lib\security目录下覆盖原来的文件
- 如果安装了JDK，将两个jar文件放到%JDK_HOME%\jre\lib\security目录下覆盖原来文件

原文链接：解密微信退款结果通知中的加密信息 req_info

实践：

以JDK8为例，系统为WIN10，替换上述security文件夹下\policy\limited文件夹和\policy\unlimite文件夹里面的local_policy.jar和US_export_policy.jar这两个文件。

若是在服务器上，则只有在security目录下有local_policy.jar和US_export_policy.jar，替换即可

解密

```
import org.bouncycastle.jce.provider.BouncyCastleProvider;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.security.Security;

public class AESUtil {

    private static Prop prop = DataDictionary.getProp();

    /**
     * 密钥算法
     */
    private static final String ALGORITHM = "AES";
    /**
     * 加解密算法/工作模式/填充方式
     */
    private static final String ALGORITHM_MODE_PADDING = "AES/ECB/PKCS7Padding";
    /**
     * 生成key
     */
    //微信支付API密钥设置路径：微信商户平台(pay.weixin.qq.com)-->账户设置-->API安全-->密设置
    private static String paySign = "微信支付API密钥";
    //对商户key做md5，得到32位小写key*
    private static SecretKeySpec key = new SecretKeySpec(MD5Util.MD5Encode(paySign, "UTF8").toLowerCase().getBytes(), ALGORITHM);

    static {

    }

    /**
     * AES加密
     *
     * @param data
     * @return
     * @throws Exception
     */
    public static String encryptData(String data) throws Exception {
        Security.addProvider(new BouncyCastleProvider());
        // 创建密码器
        Cipher cipher = Cipher.getInstance(ALGORITHM_MODE_PADDING, "BC");
        // 初始化
        cipher.init(Cipher.ENCRYPT_MODE, key);
```

```
    return Base64Util.encode(cipher.doFinal(data.getBytes()));
}

/**
 * AES解密
 *
 *（1）对加密串A做base64解码，得到加密串B
 *（2）用key*对加密串B做AES-256-ECB解密（PKCS7Padding）
 * @param base64Data
 * @return
 * @throws Exception
 */
public static String decryptData(String base64Data) throws Exception {
    Security.addProvider(new BouncyCastleProvider());
    Cipher cipher = Cipher.getInstance(ALGORITHM_MODE_PADDING, "BC");
    cipher.init(Cipher.DECRYPT_MODE, key);
    return new String(cipher.doFinal(Base64Util.decode(base64Data)));
}

public static void main(String[] args) throws Exception {
    String A = "微信返回的加密信息req_info";
    System.out.println(AESUtil.decryptData(A));
}
```