本文教你如何用Python与Java对文件进行不对称加密，并且Python与Java共用一套密钥，可以相互加解密对方的密文。本文仅作技术交流，请不要用于任何违法用途。

一、引言

最近有个项目需要做文件加密，并且要求python与java能够相互加解密。对于文件加密，我第一时间想到了比特币勒索病毒。于是收集了相关的信息，参考瑞星官网的《又一使用.net开发的勒索病毒出现——Prodecryptor勒索病毒》这篇文章，大致了解到程序先使用RSA对AES算法的密钥进行加密，然后使用AES算法对文件进行加密。

二、RSA与AES简介

RSA加密算法属于非对称加密算法，AES加密算法属于对称加密算法。这里我们不聊RSA与AES算法的具体实现，我们先聊下对称加密与非对称基本情况。

1、对称加密速度快，但是只有一个密钥进行加密和解密。当你的程序加密时，别人能反编译你的程序获取密钥，导致密钥泄露。

2、非对称加密速度慢，在加密中采用两个密钥，使用公钥进行加密，私钥进行解密。在程序中使用公钥进行加密，别人即使拿到了你的公钥也无法对文件进行解密。但是每次加密有长度限制，如果加密信息较多,需分段加解密(不建议对大量信息rsa加密,效率低效)。

因此在比特币勒索病毒程序中加密过程如下：1读取文件字节数组（加密算法都是对字节数组进行加密）2创建一个随机AES密钥，作为session密钥。3使用RSA算法对AES的session密钥加密覆盖写入文件。4然后使用AES的session密钥对文件内容进行加密写入文件。

解密过程:1读取加密后的文件字节数组，2使用私钥对AES的密钥进行解密，3使用AES对文件内容解密，4将解密后的文件内容覆盖写入文件中。

三、python与java交互遇到的一些问题

根据网上的例子，如果单独使用python或者java按上述方式进行文件加解密都很容易找到实现方案，但是如果python和java进行交互可能会出现一些问题。由于我学艺不精，采用了一些投机取巧的方法。

在java代码中我使用的是javax.crypto.Cipher模块。python中使用的是Crypto.Cipher的AES模块，以及RSA模块（也可以直接使用Crypto.Cipher的RSA模块）

问题1 Python与Java生成的RSA密钥无法相互导入

JAVA的RSAKeyPairGenerator并没有公开，因此我无法确定JAVA是如何导入相关的key。根据网上的一些方法，踩坑两小时依旧无果。最后解决方法：在对JAVA代码生成密钥debug过程中找到相应的n ，e ，d ，p ，q参数的值，也可以使用反射的方法找到这些值（私有属性，无法直接获取）。在python的rsa.newkeys函数源码中，发现可以通过（PublicKey(n, e), PrivateKey(n, e, d, p, q))这种方式导入相应的key。在Crypto.PublicKey的RSA模块中可以根据RSA.generate函数的源码找到RsaKey(n=n, e=e, d=d, p=p, q=q, u=u)这段代码导入相应的key。通过这种方式就可以python与Java使用同一套密钥了。

问题2 Python使用AES加密后的内容，python可以解密但是Java无法解密。
后发现是数据进行padding时两边不一致导致的，后将函数改为以下得到解决

```
# ?????padding = lambda s: s + (16 - len(s) % 16) * chr(16 -
 len(s) % 16).encode()
```

四、python相关代码

以下为python加密模块内容，相应的密钥是通过【内容三】中的问题1获取得到。在实际项目中应该只保留公钥部分，私钥及解密部分应该剔除，下面java相关内容中也是如此。

```
"""??????"""import rsafrom rsa.key import PublicKey, Private
Keyfrom Crypto.Random import get_random_bytesfrom Crypto.Cip
her import AESn = 125720733811994291169610359480915027242498
332835983573581162897380161904809997273335081454438575169422
082365180940876618145332064877333466150617673634912674691272
686174191003876154229771482350657467474462371208540453058063
685991322556840489981457202800248228247442801463255542794487
793252246277195836743728807e = 65537d = 191371292629881601035
575582437121142435130740930646384943553733986366406188634401
922437288699196269132775282283977999986917585984698968195955
239969973936686417032907212206649527011774897714725474886031
049229542718237754549863294574561095602754592205273482485305
289414428603551207636838864047226576028764734081p = 12485487
```

```
27443315588285592033162211686213440713521116411350723978660598965538875012059855626354430844476140779617874128869968978882850298021789066764500183 3q = 10069349401319384440381523925066814369492626716870156920501823163204449469801788824691171814613081997769220084438138226773387889070138894859614591998248079(pubkey, privkey) = (PublicKey(n, e), PrivateKey(n, e, d, p, q))def rsa_encrypt_file(file_path):    """??????"""    with open(file_path, 'rb') as f:        data = f.read()    with open(file_path, 'wb') as out_file:        session_key = get_random_bytes(16)        # ??????????AES??        cipher_session = rsa.encrypt(session_key, pubkey)        # aes??????        out_file.write(cipher_session)        # aes??    mode = AES.MODE_ECB        cryptos = AES.new(session_key, mode)        # ?????        padding = lambda s: s + (16 - len(s) % 16) * chr(16 - len(s) % 16).encode()        # AES???????        cipher_text = cryptos.encrypt(padding(data))        out_file.write(cipher_text)def rsa_decrypt_file(file_path):    """??????"""    with open(file_path, 'rb') as f:        # ?????passphrase, ????????rsa????????????        # ??rsa???128?????128??aes??        enc_session_key, cipher_text = [f.read(x) for x in (128, -1)]        # rsa??aes??    session_key = rsa.decrypt(enc_session_key, privkey)    # aes???????        cipher_aes = AES.new(session_key, AES.MODE_ECB)        data = cipher_aes.decrypt(cipher_text)    '''        ??????????????????????????16??????        ??????lambda s: s + (16 - len(s) % 16) * chr(16 - len(s) % 16).encode()????        ??????????c????16 - len(s) % 16        '''    c = data[-1]        index = 0        if c < 16:        for i, d in enumerate(data[::-1]):            if d == c:                index = i            else:                break        data = data[:- 1 - index]    with open(file_path, 'wb') as f:        f.write(data)if __name__ == '__main__':    rsa_encrypt_file(r'd:\1.txt')    rsa_decrypt_file(r'd:\1.txt')
```

四、java相关代码

最近写python代码写得比较多，函数及属性的命名都没有按照Java的驼峰命名方式，大家随意看看哈。

main.java

```
import java.io.File;import java.io.FileInputStream;import ja
va.io.FileOutputStream;public class Main { public static voi
d main(String[] args) throws Exception {  rsa_encrypt_file("
d:\\1.txt");  rsa_decrypt_file("d:\\1.txt"); } /**  * ??????
  * @param original ???????  * @param start ??????????  * @p
aram end ??????????  * @return  */ public static byte[] copy
_array(byte[] original, int start, int end) {  int newLength
 = end - start;  byte[] copy = new byte[newLength];  System.
arraycopy(original, start, copy, 0, Math.min(original.length
, newLength));  return copy; } /**  * rsa?????????aes???????
?  *   * @param file_path  * @throws Exception  */ public st
atic void rsa_decrypt_file(String file_path) throws Exceptio
n {  // ????  File f = new File(file_path);  int length = (i
nt) f.length();  byte[] data = new byte[length];  FileInputS
tream fis = new FileInputStream(f);  fis.read(data);  fis.cl
ose();  // ??  byte[] enc_session_key = copy_array(data, 0,
128);  byte[] session_key = RSAUtils.decrypt(enc_session_key
, RSAUtils.privateKeyString);  byte[] cipher_text = copy_arr
ay(data, 128, data.length);  byte[] text = AESUtil.aes_decry
pt(cipher_text, session_key);  // ????  FileOutputStream fos
 = new FileOutputStream(f);  fos.write(text);  fos.flush();
 fos.close(); } /**  * rsa?????????aes????????  *   * @param
 file_path  * @throws Exception  */ public static void rsa_e
ncrypt_file(String file_path) throws Exception {  // ????  F
ile f = new File(file_path);  int length = (int) f.length();
  byte[] data = new byte[length];  FileInputStream fis = new
 FileInputStream(f);  fis.read(data);  fis.close();  // aes?
????  byte[] session_key = AESUtil.create_aes_Key();  // ??r
sa?aes?????  byte[] cipher_session = RSAUtils.encrypt(sessio
n_key, RSAUtils.publicKeyString);  // ??aes?????????  byte[]
 cipher_data = AESUtil.aes_encrypt(data, session_key);  // ?
??????  FileOutputStream fos = new FileOutputStream(f);  fos
.write(cipher_session);  fos.write(cipher_data);  fos.flush(
);  fos.close(); }}
```

AESUtil.java

```
import java.security.Key;import java.security.NoSuchAlgorithmException;import javax.crypto.Cipher;import javax.crypto.KeyGenerator;import javax.crypto.SecretKey;import javax.crypto.spec.SecretKeySpec;public class AESUtil { /**  * ??128?????  *  * @return  * @throws NoSuchAlgorithmException  */ public static byte[] create_aes_Key() throws NoSuchAlgorithmException {  // ??key  KeyGenerator keyGenerator;  // ????????AES??,??????  keyGenerator = KeyGenerator.getInstance("AES");  // ????128?????,?????????  keyGenerator.init(128);  // ????????  SecretKey secretKey = keyGenerator.generateKey();  // ????????????  byte[] keyBytes = secretKey.getEncoded();  // key??,????????AES??  // Key key = new SecretKeySpec(keyBytes, "AES");  return keyBytes; } /**  * AES??  *  * @param  cipherText ??????byte??  * @param key_byte  ?????  */ public static byte[] aes_decrypt(byte[] cipherText, byte[] key_byte) {  Cipher cipher;  byte[] result = null;  try {  Key key = new SecretKeySpec(key_byte, "AES");  cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");  // ????????????(Encrypt_mode)????(Decrypt_mode)????????????KEY  cipher.init(Cipher.DECRYPT_MODE, key);  result = cipher.doFinal(cipherText);  } catch (Exception e) {  e.printStackTrace();  }  return result; } /**  * AES??  *  * @param context ??????? KEYSTR ?????  * @return  */ public static byte[] aes_encrypt(byte[] context, byte[] key_byte) {  try {  Key key = new SecretKeySpec(key_byte, "AES");  Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");  cipher.init(Cipher.ENCRYPT_MODE, key);  // ????????????????  return cipher.doFinal(context);  } catch (Exception e) {  e.printStackTrace();  return null;  } }}
```

RSAUtils.java

此处密钥是通过【内容三】中的问题1获取得到的。生成的密钥步骤可以参考genKeyPair函数

```
import javax.crypto.Cipher;import java.security.KeyFactory;import java.security.KeyPair;import java.security.KeyPairGenerator;import java.security.NoSuchAlgorithmException;import java.security.SecureRandom;import java.security.interfaces.RS
```

APrivateKey;import java.security.interfaces.RSAPublicKey;imp
ort java.security.spec.PKCS8EncodedKeySpec;import java.secur
ity.spec.X509EncodedKeySpec;import java.util.Base64;public c
lass RSAUtils { private static Base64.Decoder decoder = Base
64.getDecoder(); private static Base64.Encoder encoder = Bas
e64.getEncoder(); // ?????????? public static String publicK
eyString = "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCzCEKyIaoS
OCd+8XG/u6X9fGGlgmqygZPAWAYpSaPebX4kUm2yloxLdTAwYbCQmMhcgyOv
xdo9H9Qjc/uw1SY43mvIAqXaRNQ9FYbzMCcV167ebjJF4xFjPICf5bQqBh4m
t5vuf0CM1lpZazI7rsI2R5/pdVwmVXKFEVmqpuu+pwIDAQAB"; public st
atic String privateKeyString = "MIICdwIBADANBgkqhkiG9w0BAQEF
AASCAmEwggJdAgEAAoGBALMIQrIhqhI4J37xcb+7pf18YaWCarKBk8BYBilJ
o95tfiRSbbKWjEt1MDBhsJCYyFyDI6/F2j0f1CNz+7DVJjjea8gCpdpE1D0V
hvMwJxXXrt5uMkXjEWM8gJ/ltCoGHia3m+5/QIzWWllrMjuuwjZHn+l1XCZV
coURWaqm676nAgMBAAECgYAbQI6mfulcjJ+2expNjUrfIyfaAdgsA/1xsfR+
JG+FVDV3YfTA0pnYgqYrNzOhTyBwtKWiBAQMeePY4bbWXBvNGsCSd7pwP4Io
2B24fm4yKSIUbjJKx2jQMbLn+kvNu9Tw508ogmEfnHzUmVy0h2ePN+6hTUCZ
7jjaNFlK2oACgQJBAO5jymEpoTdqFluIt2ETc6ElW9yPg1IrIJNT2QSPMS1i
2xd/BmPP9PQMcV4hHv7knLFeLAjVaf0QFJ0SetlqYGkCQQDAQfRiii7xbt0s
YMIrSWe4ygjSvxC9Job1dtgyI1Q2EjfO6wZzd+7Iu+pbC2sA2fCk40YMmxSm
vCQoOuO/RESPAkEA1IjZfOi9mAcYKcFpJL5P38LL9IdqoA5dO5yMpjj3siwp
gvg3/TMBg7e4NyC2Xq/5V1TLU5DZrsnwZt1782yYyQJBAL+4hcZGWm20yqZY
jwivmNlzz7ypgD2/z9G0g//rryyEmlajlLlNHjfa/OdxyXD95LXpVo93ju5+
q+faYgb4Qw0CQG6d0blzJS9qmnkP61Q49bBfiOPLF5MF9T+VyXe7zyjGKdrn
T6WUucGTjjdVW1FX1mkMBtUdu9VPnbGiYzvoyuM="; public static voi
d main(String[] args) throws Exception { genKeyPair(); //
?????// String message = "df723820";// String messageEn =
encrypt(message, publicKeyString);// System.out.println(mes
sage + "\t????????:" + messageEn);// String messageDe = dec
rypt(messageEn, privateKeyString);// System.out.println("??
??????:" + messageDe); } /** * ??????? * * @throws NoSuch
AlgorithmException */ public static void genKeyPair() throw
s NoSuchAlgorithmException { // KeyPairGenerator???????????
???RSA?????? KeyPairGenerator keyPairGen = KeyPairGenerator
.getInstance("RSA"); // ??????????????96-1024? keyPairGen
.initialize(1024, new SecureRandom()); // ???????????keyPai
r? KeyPair keyPair = keyPairGen.generateKeyPair(); RSAPriv
ateKey privateKey = (RSAPrivateKey) keyPair.getPrivate(); //
 ???? RSAPublicKey publicKey = (RSAPublicKey) keyPair.getPu

blic(); // ???? System.out.println(publicKey.getModulus()); System.out.println(publicKey.getPublicExponent()); publicKeyString = new String(Base64.getEncoder().encodeToString(publicKey.getEncoded())); // ??????? privateKeyString = new String(Base64.getEncoder().encodeToString(privateKey.getEncoded())); System.out.println("??????" + publicKeyString); System.out.println("??????" + privateKeyString); } /** * RSA???? * * @param str   ***** * @param publicKey ?? * @return ?? * @throws Exception ?????????? */ public static String encrypt(String str, String publicKey) throws Exception { // base64***** byte[] decoded = decoder.decode(publicKey); RSAPublicKey pubKey = (RSAPublicKey) KeyFactory.getInstance("RSA")   .generatePublic(new X509EncodedKeySpec(decoded)); // RSA?? Cipher cipher = Cipher.getInstance("RSA"); cipher.init(Cipher.ENCRYPT_MODE, pubKey); String outStr = encoder.encodeToString(cipher.doFinal(str.getBytes("UTF-8"))); return outStr; } /** * RSA???? * * @param str   ***** * @param privateKey ?? * @return ?? * @throws Exception ?????????? */ public static String decrypt(String str, String privateKey) throws Exception { // 64?????????? byte[] inputByte = decoder.decode(str.getBytes("UTF-8")); // base64***** byte[] decoded = decoder.decode(privateKey); RSAPrivateKey priKey = (RSAPrivateKey) KeyFactory.getInstance("RSA")   .generatePrivate(new PKCS8EncodedKeySpec(decoded)); // RSA?? Cipher cipher = Cipher.getInstance("RSA"); cipher.init(Cipher.DECRYPT_MODE, priKey); String outStr = new String(cipher.doFinal(inputByte)); return outStr; } public static byte[] decrypt(byte[] inputByte, String privateKey) throws Exception { // base64***** byte[] decoded = decoder.decode(privateKey); // base64***** RSAPrivateKey priKey = (RSAPrivateKey) KeyFactory.getInstance("RSA")   .generatePrivate(new PKCS8EncodedKeySpec(decoded)); // RSA?? Cipher cipher = Cipher.getInstance("RSA"); cipher.init(Cipher.DECRYPT_MODE, priKey); return cipher.doFinal(inputByte); } public static byte[] encrypt(byte[] inputByte, String publicKey) throws Exception { // base64***** byte[] decoded = decoder.decode(publicKey); RSAPublicKey pubKey = (RSAPublicKey) KeyFactory.getInstance("RSA")   .generatePublic(new X509EncodedKeySpec(decoded)); // RSA?? Cipher cipher = Cipher.getI

```
nstance("RSA");  cipher.init(Cipher.ENCRYPT_MODE, pubKey);
return cipher.doFinal(inputByte); }}
```