实验5

基于CrypTool软件的RSA算法加解密实验

5.1 实验目的

（1）理解非对称密码算法的基本思想和特点；

（2）基于 CrypTool 软件，通过实际例子深入理解典型非对称密码算法 RSA 的加解密原理，加深对非对称/公钥算法的理解与认识；

（3）体会理论与实践的关系。

5.2 实验任务

基于 CrypTool 软件，生成用户自己的公钥私钥对，应用 RSA 算法加密明文，并对比算法的运算时间。

5.3 实验环境

5.3.1 硬件环境

Windows 操作系统的计算机 1 台。

5.3.2 软件环境

CrypTool 软件。

5.4 实验学时与要求

学时：2 学时。

要求：独立完成实验任务，撰写实验报告。

5.5　理论提示

5.5.1　RSA 算法

RSA 公钥密码算法是 1977 年由 Ron Rivest、Adi Shamirh 和 LenAdleman 在美国麻省理工学院开发的。RSA 取名来自三位开发者的名字。RSA 是目前最有影响力的公钥加密算法，它能够抵抗目前已知的所有密码攻击，已被 ISO 推荐为公钥数据加密标准。所谓的公开密钥密码体制就是使用不同的加密密钥与解密密钥，是一种“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。

RSA 算法基于一个十分简单的数论事实：将两个大素数相乘十分容易，但要对其乘积进行因式分解却极其困难，因此可以将乘积公开作为加密密钥。

RSA 算法是一种非对称密码算法。所谓非对称，就是指该算法需要一对密钥，使用其中一个加密，则需要用另一个才能解密。加密密钥也称为公开密钥，所有用户的公开密钥都可以对外公开，被所有用户访问。然而，每个用户的解密密钥由用户保存并严格保密。解密密钥也称为私有密钥。

实体 B 加密明文消息 M，将密文在公开信道上传送给实体 A，实体 A 接到密文后对其解密。采用 RSA 算法的具体过程如下。

1. 公钥私钥的生成

每个通信实体都需要自己的一对 RSA 的密钥。具体步骤如下：

(1) 随机选择两个大素数 p 和 q，计算 $n=p\times q$；

(2) 根据欧拉函数计算 $r=(p-1)\times(q-1)$；

(3) 选择一个与 r 互质的整数 e，$e<r$；

(4) 求得 e 关于模 r 的模反元素，命名为 d，满足 $e\times d=1 \bmod r$。

(5) p、q 不再需要，可以销毁。

(6) (n,e)即为公钥，(n,d)即为私钥。

公钥(n,e)可以公开发布，而私钥(n,d)必须严格保管。

注意：e 和 d 分别被称为公钥指数/模值(modulus)和私钥指数/模值。“密钥长度”一般指模值的位长度。为了保证 RSA 的安全性(防止破解)，目前主流的密钥长度都是 1024 位以上，如 1024、2048、3072、4096 等公钥指数 e 可以随意选，但目前行业上普遍选用 65537 (十六进制 0x10001，该值是除了 1、3、5、17、257 之外的最小素数)，选择较小的公钥指数，加密时间就会变短；相应地，计算得到的私钥指数 d 则会更大，解密时间会变长。这种方式比较符合大多数典型应用的需求。1024 位长的模值密钥是一个较大的数字，换成十进制数表达式，如下所示。

```
2^1024 = 179769313486231590772930519078902473361797697894230657273430081157732675805500
963132708477322407536021120113879871393357658789768814416622492847430639474124377767789
342486548527630221960124609411945308295208500576883815068234246288147391311054082723716
335051068458629823994724593847971630483535632962422413721 6
```

以下是实际应用中的一个 1024 位公钥-私钥对的示例：

公钥(n,e)为

(92040937025083980114538037804534268612981293602870457611988248538976180648735579272757579573088724190319748632610086665775981559508595751014422290955541625572601174257174505870980479486337170930025402752121033334523093496428774445280659798459305405393798121701680035096537995764758495172606032836456620308447, 65537)

私钥(n,d)为

(92040937025083980114538037804534268612981293602870457611988248538976180648735579272757579573088724190319748632610086665775981559508595751014422290955541625572601174257174505870980479486337170930025402752121033334523093496428774445280659798459305405393798121701680035096537995764758495172606032836456620308447,22990221387927808024093072292906693580641527324701914813132240239605720085139713943193029549894905396883200102473825758254921923938473113571052875002215062020950422905387668296071326844919479266104757501461061878782780797278743292486902134606212493546888161451230031765354440774551400814722753904279175971633)

如果将其转换为符合公钥加密标准(Public Key Cryptography Standards,PKCS)的PKCS1规范要求的格式,如下所示:

公钥为

```
-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBAIMSFcs5eGRvw8maaq7gikmOo4y22oA/AVbnWRmubg8tJNJr2lxJXHJ3n5VgCjubZRICFXKZ1vkSvZEAr0
zH7TQHfBqrS2Adw8YNIOhuDNLo32Te6sQCaE59rZ7KqajnNJcWmAMwDrCTbfHXsxSNjYxO+AzvmSYUlMoj/iNlwT
ffAgMBAAE=
-----END RSA PUBLIC KEY-----
```

私钥为

```
-----BEGIN RSA PRIVATE KEY-----
MIICYAIBAAKBgQCDEhXLOXhkb8PJmmqu4IpJjqOMttqAPwFW51kZrm4PLSTSa9pcSVxyd5+VYAo7m2USAhVymdb5
Er2RAK9Mx+00B3waq0tgHcPGDSDobgzS6N9k3urEAmhOfa2eyqmo5zSXFpgDMA6wk23x17MUjY2MTvgM75kmFJTK
I/4jZcE33wIDAQABAoGAIL05uGOIkP3h18+8aiYoJKt+ar2Z4oLaYMy00tdhImVSV0UdbAPfFbCPqg4tQCpWmqL
unIuUyO5Hb5rOA82FAfQu+gApt4H/R4ENucoYXIlyuAiBdKUoyGZj2zY13OPvQxgcFRM/ltPG20XPhVVCorFqel+
7YktV9Y8IOvO2QzECRQCsJqautY518fE3Lt+Ylb7SVEO2JZ/W8WjAgmGRXPiYcYhE1MOagCmneiRkknW857FzaQa
O3XL3o3tjNX+UoVdpXoHkvQI9AMLpKhX6Wt7uCbz/laCNOHRz7ikeaUvg8K4+ewdBV/v99BU9GD6QIUk8O1Qaf6
t+ikxU9CjOOBdCtWCOywJEGZD/7jBugNu2rDBc6qdKmKgFClcaHuzpVrXGUbVSAinSSA6lIhvok8PTBdCuhaiPNQ
mi/LKssVRpLbmzQdqYRcDQX60CPEAJk0Y9RvXPh2uNHX/uZgn8vNbNoxLHCMx2wr0TfvDP8w23UOvdNtJ085kiS6f
wn0eM8dsmRc9+n1DmGQJFAIEA4wlyLZS3Whq65r130gj45qj29l3KWcCHV4h64xAseZCBv2RjwSDwRZeqPrycb3/
6M3exEkSpIwTE7+iZeTzkHP0J
-----END RSA PRIVATE KEY-----
```

PKCS1规范是公钥密码学领域的一系列标准之一,主要涉及RSA密码算法的实现。PKCS(Public Key Cryptography Standards)是由国际标准化组织(ISO)和国际电工委员会(IEC)共同制定的一套关于公钥加密技术的标准。PKCS1规范具体定义了使用RSA算法进行数字签名和数据加密的方法。

2. 明文加密算法

实体B加密明文消息M的操作如下:

(1) 得到实体A的真实公钥(n,e);

(2) 把消息M转换为整数m,要满足$0<m<n$;

(3) 计算密文,$C=m^e \bmod n$;

(4) 将密文C发送给实体A。

注意：明文消息 M 在加密前，需要先转换为二进制数组，这个二进制数组作为一个大整数 m 被加密处理。按照 RSA 算法原理，消息 M 存在长度限制，一般来说要小于密钥长度。对 1024 位密钥来说，明文最大长度要小于 1024/8＝128 字节（还要扣除 11 字节的固定填充）；对 2048 密钥来说，明文最大长度要小于 2048/8＝256 字节（还要扣除 11 字节的固定填充）。如果明文长度超过最大限制，则可以将明文消息分段进行加密处理，解密后再将分段拼接回原始明文。因此，RSA 算法一般并不适合用来加密特别长的明文，而是适合加密一些摘要、签名等短消息。

3. 密文解密

实体 A 接收到密文 C，使用自己的私钥 d 计算密文，$m \equiv C \hat{} d \mod n$。

图 5-1 描述了一个 RSA 算法加密解密的实际应用流程。

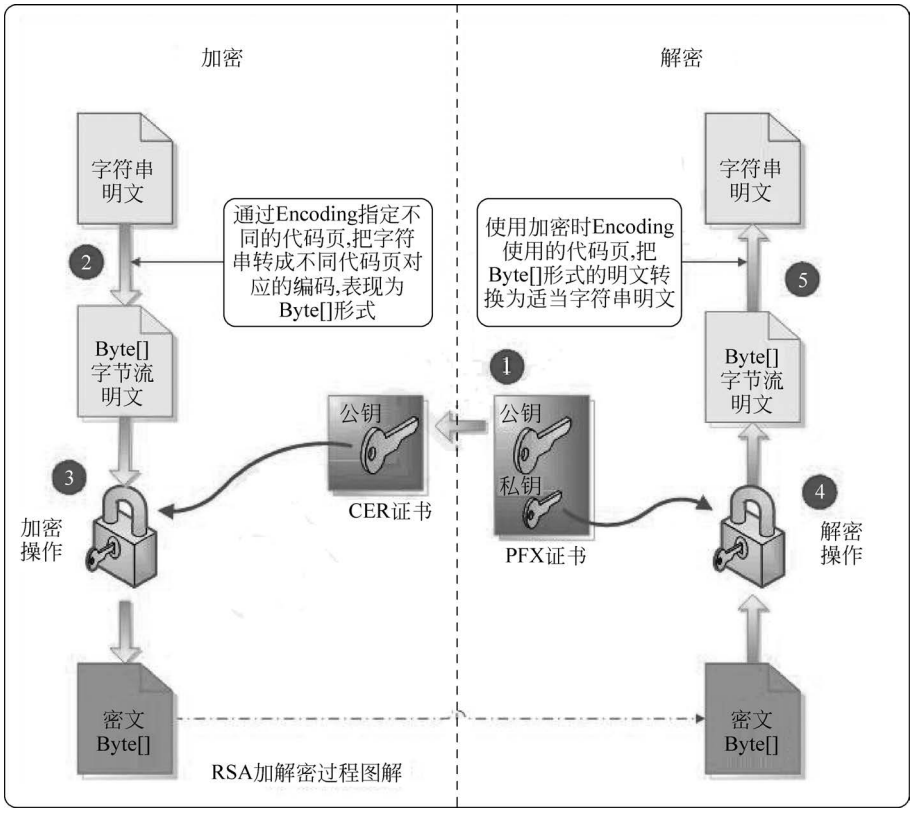

图 5-1　RSA 密码算法应用流程

从运算性能上看，公钥密码算法比对称密码加密的速度要慢，粗略地说，公钥密码算法 RSA 硬件实现比对称密码算法 DES 硬件实现的速度慢 1500 倍，而软件实现的速度要慢 100 倍。

公钥加密有以下优点：

- 大型网络中的每个用户需要的密钥数量少；
- 对管理公钥的可信第三方的信任程度要求不高而且是离线的；
- 只有私钥是保密的，而公钥只需保证它的真实性。

缺点：

- 多数公钥密码算法比对称密码算法加密的速度要慢几个数量级；
- 公钥密码算法的密钥长度比对称密码算法的密钥要长；
- 公钥密码算法没有在理论上被证明是安全的。

5.5.2 CrypTool 软件

CrypTool (CT)是一个面向 Windows 的现代电子学习程序，它将密码学和密码分析可视化。它不仅包括密码的加密和密码分析，还包括它们的基础知识和现代密码学的全部内容。

CT 包含 200 多个具有工作流的现成模板，还可以轻松地组合和执行加密函数，以自己在 CT 中创建工作流(可视化编程)。使用这种方法，复杂的过程可以很容易地可视化，从而更好地理解。通过使用矢量图形，可以自由缩放当前视图。

5.6 实验指导

5.6.1 RSA 密钥对生成

(1) 双击运行 SetupCrypTool_1_4_41_en.exe，打开 Cryptool 软件。

(2) 依次选择软件上方菜单栏 Digital Signature/PKI→PKI→ Generate/Import Keys，如图 5-2 所示。

图 5-2 启动 CrypTool 软件

(3) 在弹出的如图 5-3 所示界面中，选择生成 RSA 密钥对的基本参数，如密钥的位数；填写密钥拥有者姓名，本例为[×××][×××]；设置该密钥对的使用权限密码(必填)PIN 等，然后单击左下方第一个按钮 Generate new key pair 生成需要的密钥对。这一步由于计算机性能不同，需要花费的时间稍有不同。

(4) 密钥对生成后，弹出如图 5-4 所示界面，标识了密钥拥有者、位数等信息，即可为该用户生成一个新的 1024 位长的 RSA 密钥对，该密钥对含有 1 个公钥和 1 个配对私钥。

此时，原图 5-3 下方 Show key pair 按键将由原来的灰色不可用状态转换为可单击状态。

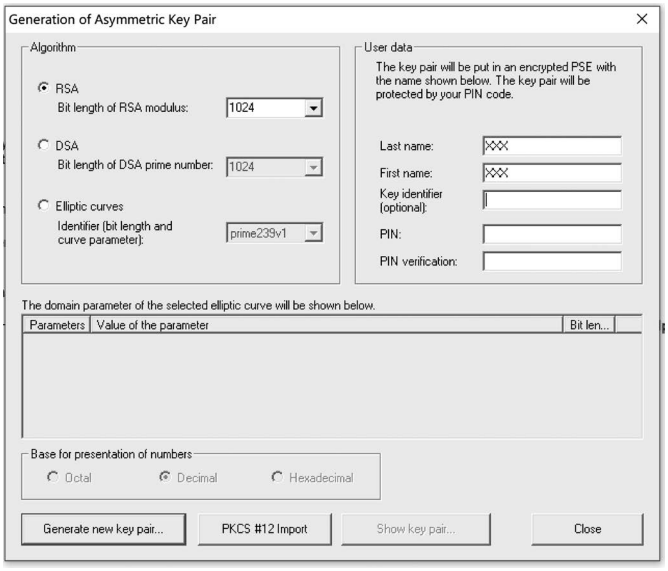

图 5-3　RSA 密钥对生成参数设置

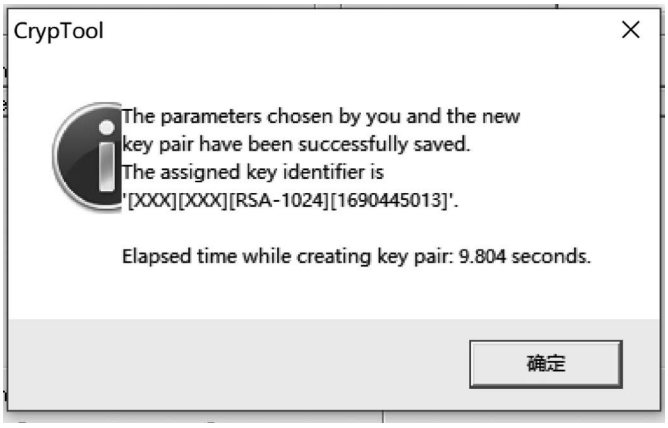

图 5-4　RSA 密钥对生成成功

(5) 单击图 5-3 下方 Show key pair 按钮，或依次单击软件上方菜单栏 Digital Signature/PKI→PKI→ Display/Export Keys，可以查看、管理和维护系统中所有可用密钥对，生成其他类型的密码编码格式(如 PKCS)，弹出如图 5-5 所示界面。选中需要查看的公钥对，单击图 5-5 界面中的 Show public parameters 按钮，弹出如图 5-6 所示界面。选择不同的公钥显示格式，可以看到不同进制的公钥$\{e,n\}$的数值。其中，Octal 表示八进制，Decimal 表示十进制(默认)，Hexadecimal 表示十六进制。

5.6.2　RSA 算法加密

1. 创建明文

依次单击 CrypTool 软件上方菜单栏 File→new，在弹出的对话框中输入需要加密的明文字符，然后单击“保存”按钮，明文将以. text 格式存储在指定位置，或者依次单击 File→open，打开一个提前创建好的. txt 文件。本例以“I love IOT security”为例，如图 5-7 所示。

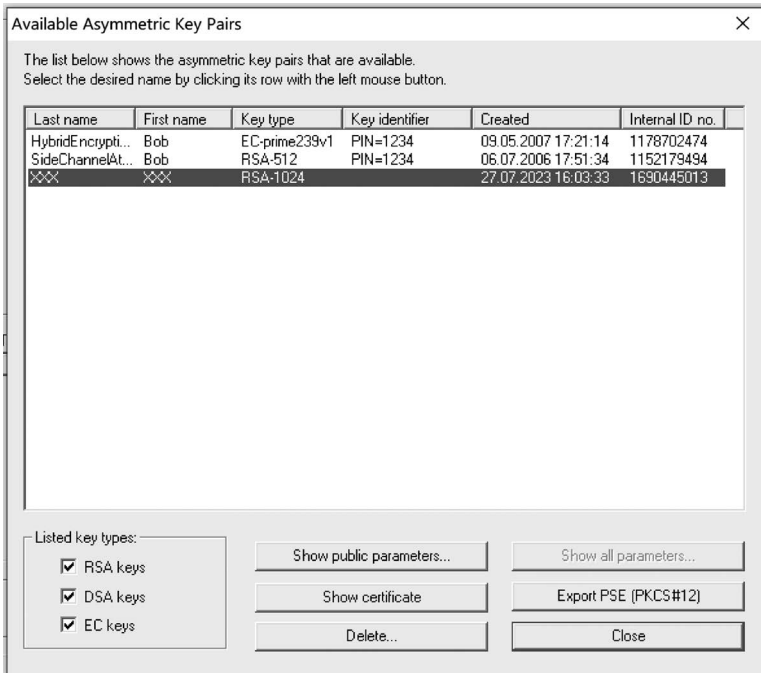

图 5-5　查看生成的 RSA 密钥对

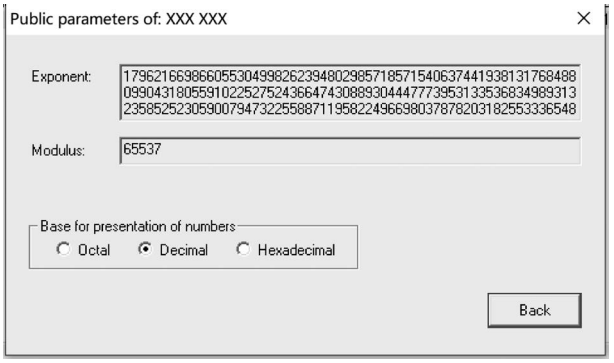

图 5-6　十进制的 RSA 公钥$\{e, n\}$

图 5-7　创建明文

2. 选择用于加密的 RSA 公钥

依次单击 CrypTool 软件上方菜单栏 Encrypt/Decrypt→Asymmetric→RSA Encryption，在如图 5-8 所示弹出的界面中选中刚才生成的公钥加密。

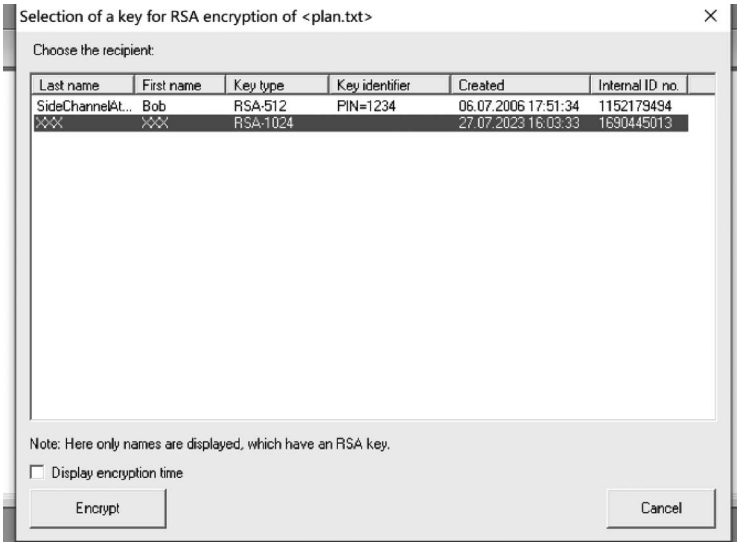

图 5-8　选择用于加密的 RSA 公钥

3. 保存密钥

单击图 5-8 左下方的 Encrypt 按钮，将弹出如图 5-9 所示的密文信息，默认情况下是十六进制。然后单击“保存”按钮，密文将以.hex 格式存储在指定位置。

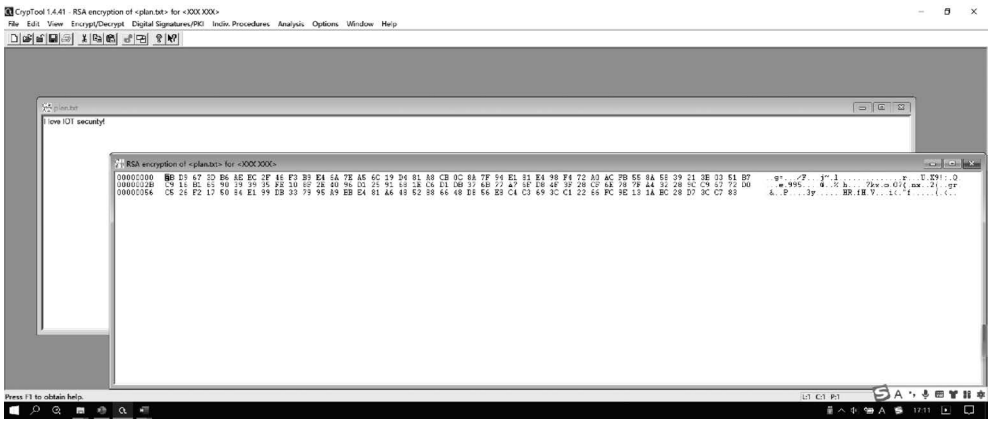

图 5-9　RSA 加密的密文信息（十六进制格式）

5.6.3　RSA 算法解密

依次单击 CrypTool 软件上方菜单栏 Encrypt/Decrypt→Asymmetric→RSA Decryption，在如图 5-10 所示弹出的界面中选中刚才生成的公钥，同时须输入公钥创建时设置的 PIN。然后单击左下方的 Decrypt 按钮，如图 5-11 所示，显示解密明文与原始明文一致，表明解密成功。

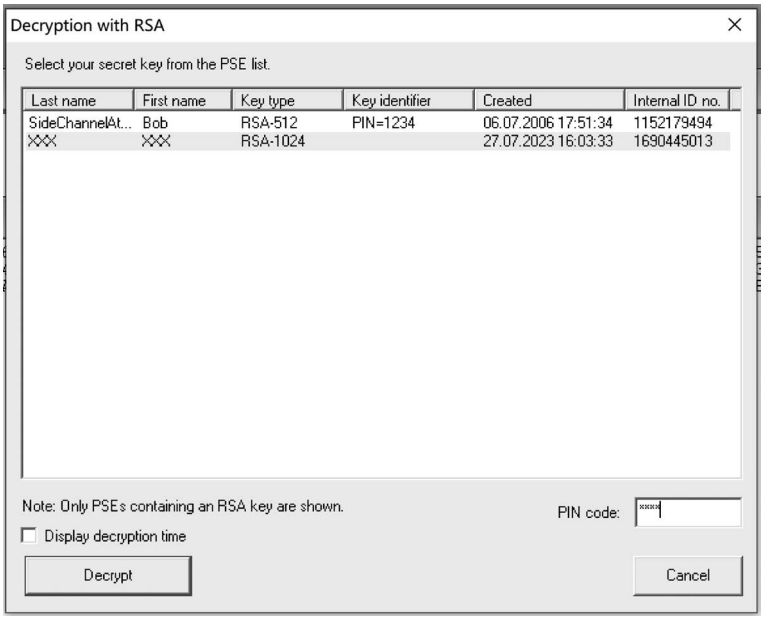

图 5-10　选择 RSA 解密的私钥信息(十六进制格式)

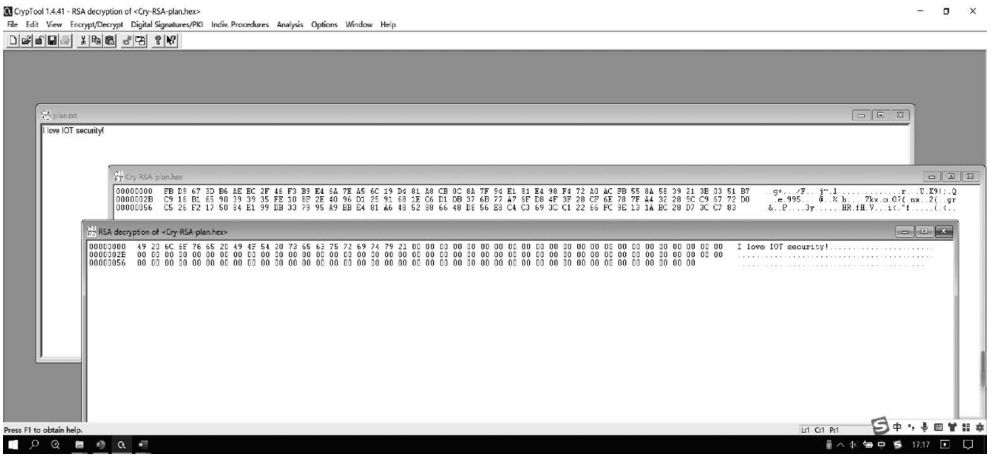

图 5-11　用私钥解密后的明文信息

5.7　注意事项

1. 熟悉 CrypTool 软件

熟悉 CrypTool 软件的界面和功能,知道如何生成密钥对,如何使用公钥加密信息,以及如何使用私钥解密信息。同时,要了解软件中的错误检测和调试工具,以便在实验过程中能够快速定位和解决问题。

2. 密钥管理

强调私钥的保密性。在实验中,私钥一旦泄露,就意味着任何人都可以解密信息,这将

严重威胁到通信的安全性。在实验中应妥善处理密钥，不要在不安全的地方存储或传输私钥。

3. 实践与理论相结合

在实验中应观察和思考理论如何转换为实际操作。例如，在实验过程中，加密和解密的速度可能比理论预期的要慢，这是因为大数运算的复杂性。同时，注重为提高效率，可以使用优化算法和硬件加速。

4. 算法的灵活运用

在实际应用中，非对称密码算法通常与对称密码算法结合使用。非对称加密用于安全地交换对称密钥，而对称加密则用于实际的数据传输，因为对称加密在速度上更优。

5. 实验记录和分析

在实验过程中详细记录每一步操作和结果，包括任何遇到的问题和解决方案。这不仅有助于他们理解实验过程，也是学习如何分析和解决问题的好方法。

6. 遵守实验规则

遵守实验室的所有规则和安全协议，包括不进行任何未经授权的实验步骤，不尝试破解他人的加密信息，以及不泄露实验中生成的任何密钥。

5.8　思考题

1. RSA 算法中，公钥和私钥间的关系是什么？

2. 简述设置公钥加密标准（Public Key Cryptography Standards，PKCS）的 PKCS1 规范的必要性。

3. 在军事物联网中，RSA 算法因其非对称加密特性被广泛应用于密钥交换和数据加密，请思考：

（1）RSA 算法如何帮助军事物联网实现安全的密钥分发？特别是在分布式网络中，如何确保公钥的安全传输和私钥的保密性？

（2）在军事物联网环境中，如何设计一个有效的密钥管理系统来管理大量的 RSA 密钥对？考虑到设备的动态加入和离开，以及密钥的定期更换需求。

（3）分析 RSA 密钥长度（如 1024 位、2048 位等）对军事物联网安全性和性能的影响，并讨论如何选择适当的密钥长度。

4. RSA 算法虽然安全，但相对于对称密码算法（如 AES 算法）来说，其计算复杂度较高，请思考：

（1）在军事物联网中，如何优化 RSA 算法的加密解密性能，以满足实时性要求？是否可以通过硬件加速、算法优化或选择合适的加密模式（如 ECB、CBC 等）来实现？

（2）对于传输大量数据的场景，是否可以将 RSA 算法与其他加密算法（如 AES）结合使用，以平衡安全性和性能？具体如何实施？

（3）分析在资源受限的军事物联网设备（如传感器、无人机等）上运行 RSA 加密的可行性，并讨论可能的解决方案。

5. 随着计算能力的提升（如量子计算机的问世），RSA 算法的安全性也受到了一定的

挑战，请探讨：

（1）RSA 算法在军事物联网中面临的主要安全威胁有哪些？如何防御这些威胁，如量子计算攻击、侧信道攻击等？

（2）考虑到 RSA 密钥的敏感性，如何确保在密钥生成、存储、传输和使用过程中的安全性？是否需要采用额外的安全措施，如物理隔离、访问控制等？

（3）分析在军事物联网中实施定期更换 RSA 密钥的必要性和实施策略，以确保长期的安全性。

6. 除了传统的加密解密和密钥交换外，RSA 算法在军事物联网中还有哪些创新的应用场景？请思考：

（1）RSA 算法在军事物联网中的创新应用。如如何利用 RSA 算法的特性来实现军事物联网中的身份认证和访问控制？具体实现方案是什么？

（2）探讨 RSA 算法在军事物联网中用于数据完整性校验的可能性，如结合数字签名技术来验证数据的真实性和完整性。

（3）分析 RSA 算法在军事物联网中与其他安全技术的集成（如区块链、安全多方计算等），并讨论这些集成如何进一步提升军事物联网的安全性和可靠性。