

Final Assignment Report
Security CMPU3034

Grade =52

Student Number: D16123932

Student Name: Zhimian Wu

Implementation of Vigenère and RSA

Vigenere cipher has several Caesar ciphers in sequence with different shift values. To encrypt the plaintext, need a keyword and repeat it until matches the length of the plain text. Then find which letter paired with the first letter of the plaintext, use the Vigenere table or Vigenere square to find the letter then use that row to encrypt the plaintext. For example, the keyword is “LEMON”, plaintext is “CIPHER”, so I got the keyword repeat “LEMONL”. Then “L” is paired with the first letter of plaintext “A”, so use row “L” and column C of the Vigenere square are used, namely “L”. In that way, I got the ciphertext “NMBVRC”.

```
public class VigenereCipher
{
    public static void main(String arg[])
    {
        String plaintext = "CIPHER";
        String keyword = "LEMON";
        encryptDecrypt(plaintext, keyword);
    }

    public static void encryptDecrypt(String plaintext, String keyword)
    {
        //Converting plaintext to char array
        char msg[] = plaintext.toCharArray();
        int msgLen = msg.length;
        int i, j;

        // Creating new char arrays
        char key[] = new char[msgLen];
        char encryptedMsg[] = new char[msgLen];
        char decryptedMsg[] = new char[msgLen];

        // Generate key
        for (i = 0, j = 0; i < msgLen; i++, j++)
        {
            if (j == keyword.length())
            {
                j = 0;
            }
            key[i] = keyword.charAt(j);
        }

        //encryption code
        for (i = 0; i < msgLen; i++)
            encryptedMsg[i] = (char)((((msg[i] + key[i]) % 26) + 'A'));

        //decryption code
        for (i = 0; i < msgLen; i++)
            decryptedMsg[i] = (char)((((encryptedMsg[i] - key[i] + 26) % 26) + 'A'));

        System.out.println("Original Message: " + plaintext);
        System.out.println("Keyword: " + keyword);
        /* String.valueOf() method converts
        char[] to String */
        System.out.println("Key: " + String.valueOf(key));
        System.out.println();
        System.out.println("Encrypted Message: " + String.valueOf(encryptedMsg));
        System.out.println();
        System.out.println("Decrypted Message: " + String.valueOf(decryptedMsg));
    }
}
```

```

F:\DT228-3\Security>java VigenereCipher
Original Message: CIPHER
Keyword: LEMON
Key: LEMONL

Encrypted Message: NMBVRC

Decrypted Message: CIPHER

F:\DT228-3\Security>_

```

Select two prime number p and q and get $n = p \cdot q$ and then randomly select the encryption key ' e ' and make ' e ' and $(p-1)(q-1)$ prime. Get ' e ' and ' n ' as public key and d is private key based on $d = e^{-1} \bmod ((p-1)(q-1))$

```

import java.util.*;
import java.math.*;

public class RSA
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in); // scan user input

        int p,q,n = 0; // two prime number p|q and n equal to p*q
        int e,d = 0; // public key exponent and private key exponent
        int z; // φ(n)

        System.out.println("Enter the number to be encrypted and decrypted");
        int msg = sc.nextInt();
        double c;
        BigInteger msgback;
        System.out.println("Enter 1st prime number p");
        p = sc.nextInt();
        System.out.println("Enter 2nd prime number q");
        q = sc.nextInt();

        n = p*q;
        z=(p-1)*(q-1);
        System.out.println("the value of z = "+z);

        for(e = 2; e < z; e++)
        {
            if(gcd(e,z)==1)
            {
                break;
            }
        }
        System.out.println("the value of e = "+e);

        for(int i = 0; i <= 9; i++)
        {
            int x = 1 + (i*z);
            if(x%e==0)
            {
                d=x/e;
                break;
            }
        }
        System.out.println("the value of d = "+d);
    }
}

```

```

        c=(Math.pow(msg,e))%n;
        System.out.println("Encrypted message is : ");
        System.out.println(c);
        BigInteger N = BigInteger.valueOf(n);
        BigInteger C = BigDecimal.valueOf(c).toBigInteger();
        msgback = (C.pow(d)).mod(N);
        System.out.println("Decrypted message is : ");
        System.out.println(msgback);
    }

    static int gcd(int e, int z)
    {
        if(e == 0)
        {
            return z;
        }
        else
        {
            return gcd(z%e, e);
        }
    }
}

```

```

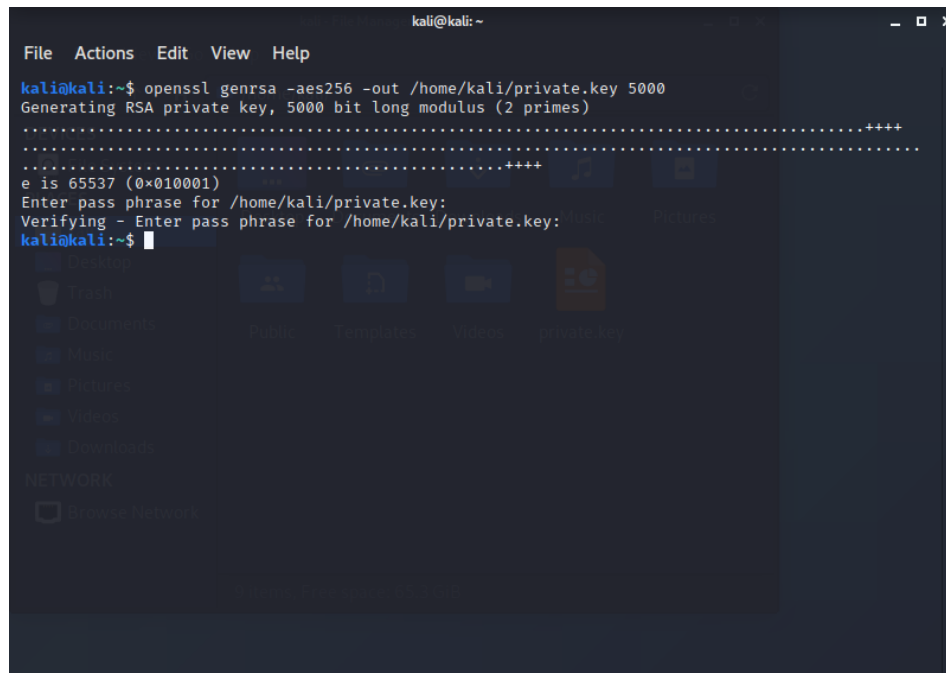
F:\DT228-3\Security>java RSA
Enter the number to be encrypted and decrypted
12
Enter 1st prime number p
5
Enter 2nd prime number q
7
the value of z = 24
the value of e = 5
the value of d = 5
Encrypted message is :
17.0
Decrypted message is :
12

```

Cryptographic Tools

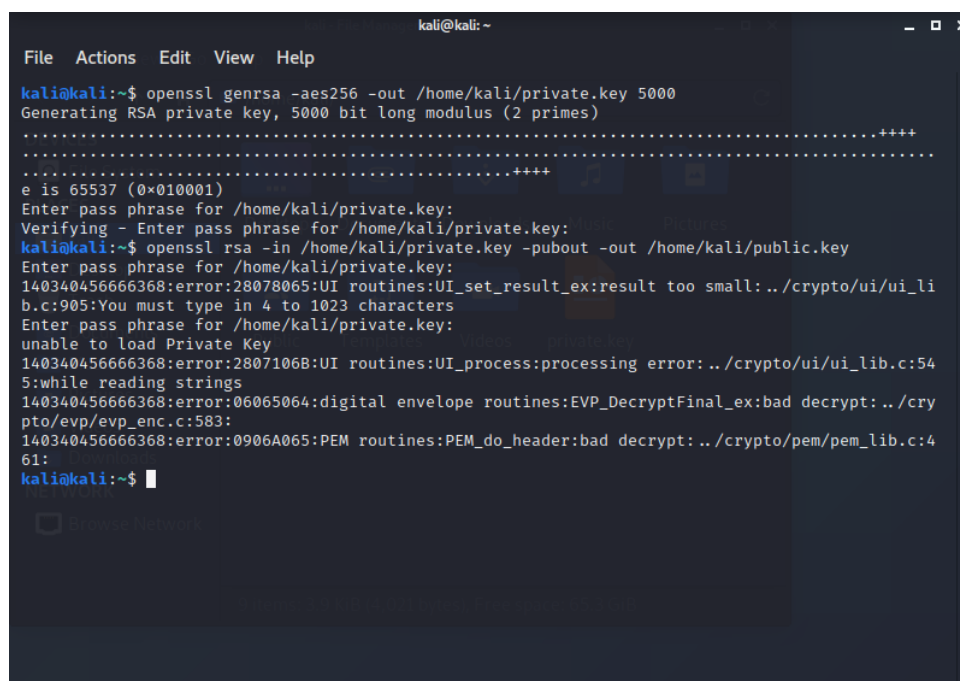
RSA

Firstly, a 5000-bit RSA key (private key) is generated and encrypted by AES256 algorithm. In the process, a password needs to be entered and set to 123456.



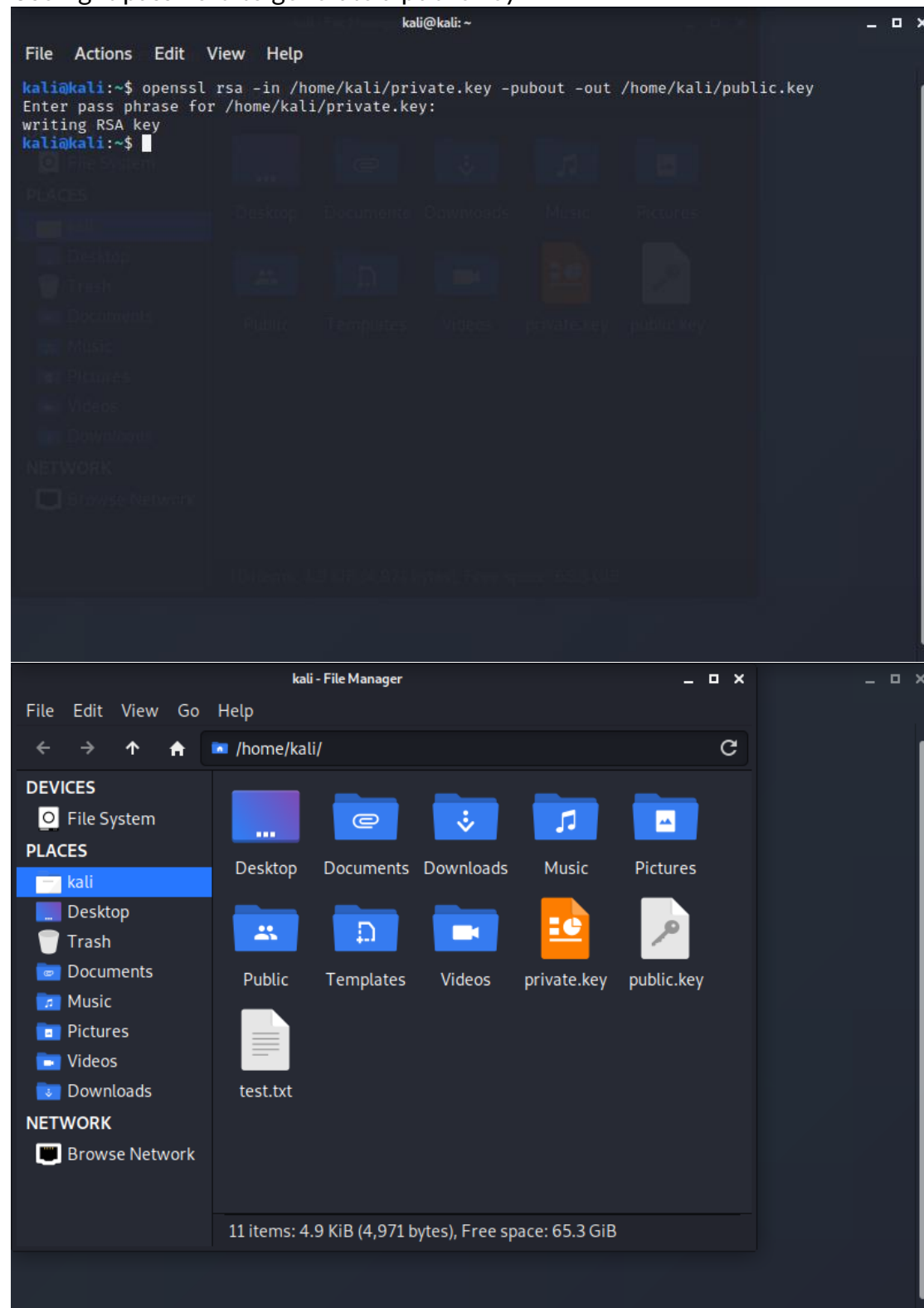
```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ openssl genrsa -aes256 -out /home/kali/private.key 5000  
Generating RSA private key, 5000 bit long modulus (2 primes)  
.....++++  
.....++++  
e is 65537 (0x010001)  
Enter pass phrase for /home/kali/private.key:  
Verifying - Enter pass phrase for /home/kali/private.key: 123456  
kali@kali:~$
```

Then try to generate a public key using private key with wrong password. First try 123 as password, then try 1234 to see if I can use wrong password to use private key. In fact, I can't generate the public key.

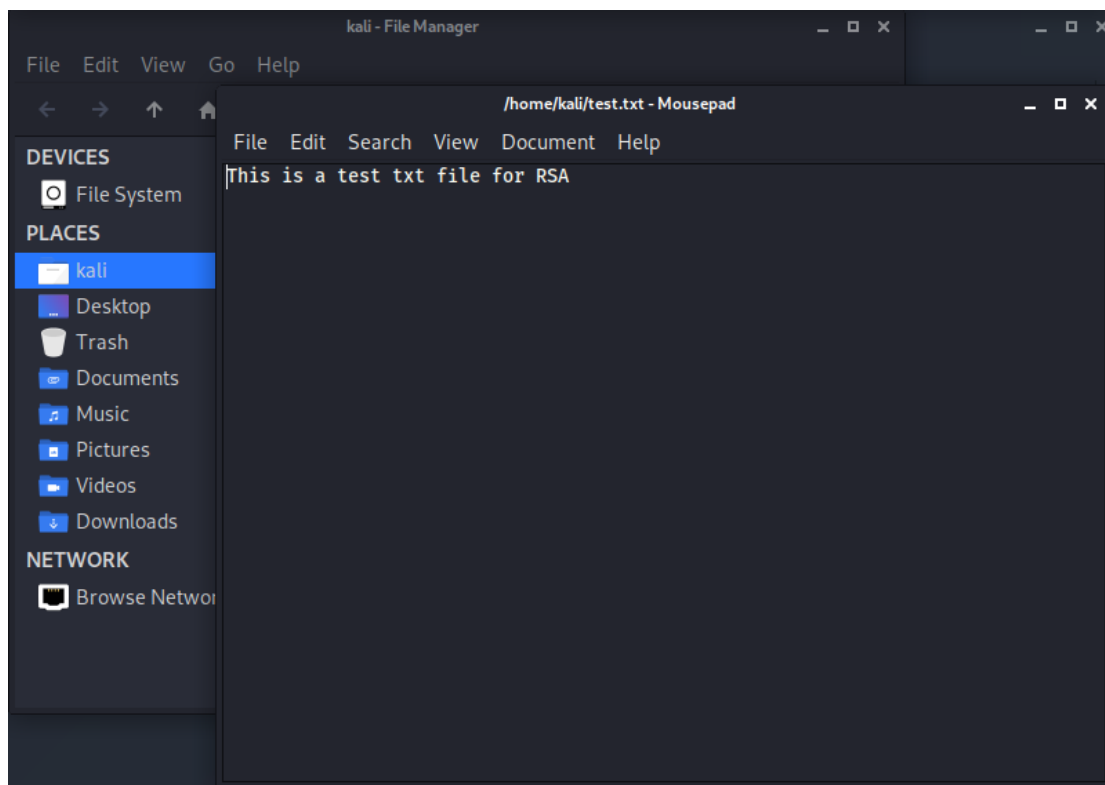


```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ openssl genrsa -aes256 -out /home/kali/private.key 5000  
Generating RSA private key, 5000 bit long modulus (2 primes)  
.....++++  
.....++++  
e is 65537 (0x010001)  
Enter pass phrase for /home/kali/private.key:  
Verifying - Enter pass phrase for /home/kali/private.key:  
kali@kali:~$ openssl rsa -in /home/kali/private.key -pubout -out /home/kali/public.key  
Enter pass phrase for /home/kali/private.key:  
140340456666368:error:28078065:UI routines:UI_set_result_ex:result too small:../crypto/ui/ui_lib.c:905:You must type in 4 to 1023 characters  
Enter pass phrase for /home/kali/private.key:  
unable to load Private Key  
140340456666368:error:2807106B:UI routines:UI_process:processing error:../crypto/ui/ui_lib.c:545:while reading strings  
140340456666368:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:../crypto/evp/evp_enc.c:583:  
140340456666368:error:0906A065:PEM routines:PEM_do_header:bad decrypt:../crypto/pem/pem_lib.c:461:  
kali@kali:~$
```

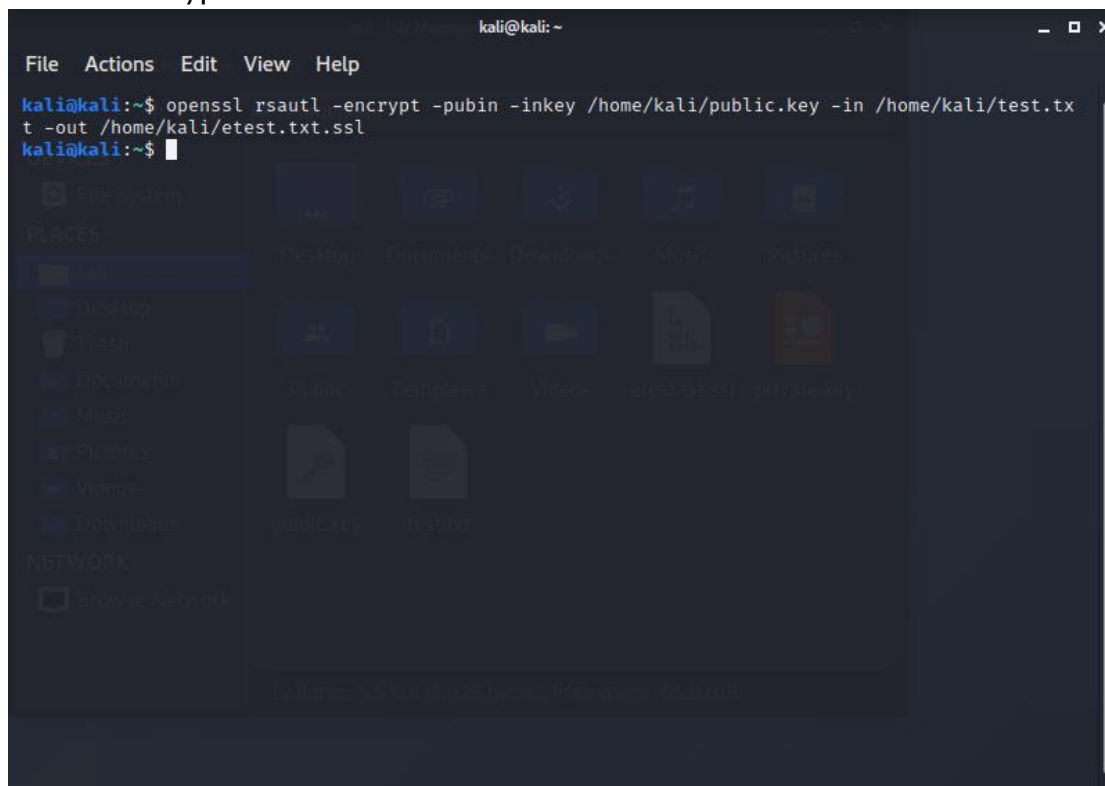
Use right password to generate a public key

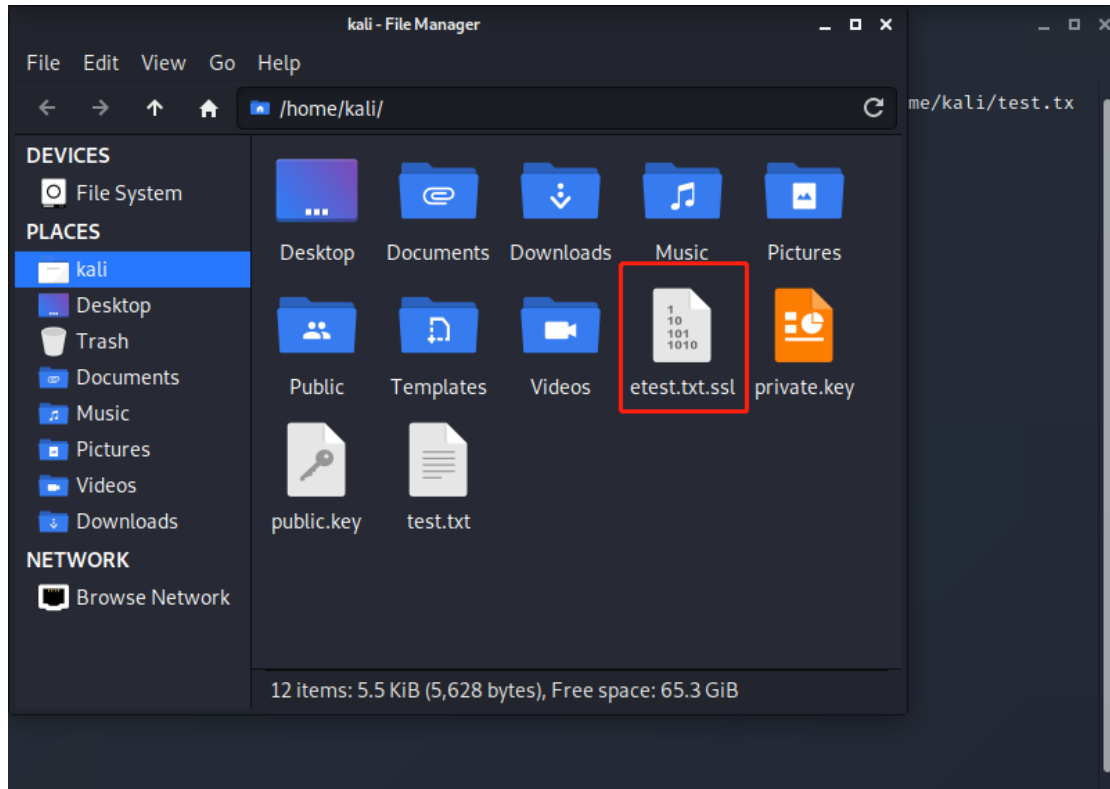


Then create a txt file for test.

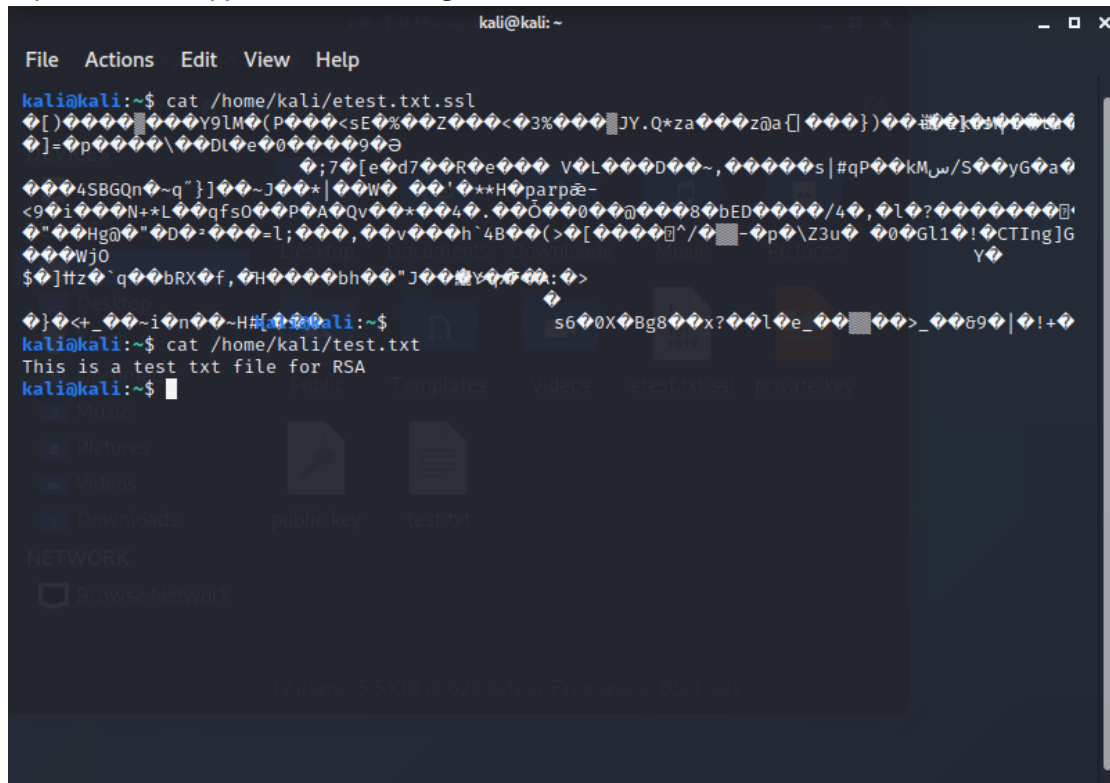


Start to encrypt the test file





Open the encrypted file and original file to view the contents, see what different.



Start decrypting the encrypted file

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ cat /home/kali/etest.txt.ssl  
[Y9LM(P<SE%Z<3%JY.Q*zaZ@a[})#k]osYCa  
]=p\DLDe09Θ  
;7[e0d7RRe V0L0D0~,,s|#qPkmس/SyG@a  
4SBGQn~q"}~J~*|W *Hparpæ-  
<9iN+*Lqfs0PQAQv*4.Ö0008bED/4,1?:  
"HgD=D=l;,,v`h`4B(>[^/-p\Z3u 0Gl1!CTIng]G  
WjO  
$]tz`q`bRXf,Hbh"JYV:A:>  
} <+_~i~n~H#s60XBg8x?le_>_69|!+<br>kali@kali:~$ cat /home/kali/test.txt  
This is a test txt file for RSA  
kali@kali:~$ openssl rsautl -decrypt -inkey /home/kali/private.key -in /home/kali/etest.txt.ssl  
-out /home/kali/dtest.txt  
Enter pass phrase for /home/kali/private.key:  
kali@kali:~$
```

Compare content with original file

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ cat /home/kali/etest.txt.ssl  
[Y9LM(P<SE%Z<3%JY.Q*zaZ@a[})#k]osYCa  
]=p\DLDe09Θ  
;7[e0d7RRe V0L0D0~,,s|#qPkmس/SyG@a  
4SBGQn~q"}~J~*|W *Hparpæ-  
<9iN+*Lqfs0PQAQv*4.Ö0008bED/4,1?:  
"HgD=D=l;,,v`h`4B(>[^/-p\Z3u 0Gl1!CTIng]G-U  
WjO  
$]tz`q`bRXf,Hbh"JYV:A:>  
} <+_~i~n~H#s60XBg8x?le_>_69|!+96  
kali@kali:~$ cat /home/kali/test.txt  
This is a test txt file for RSA  
kali@kali:~$ openssl rsautl -decrypt -inkey /home/kali/private.key -in /home/kali/etest.txt.ssl  
-out /home/kali/dtest.txt  
Enter pass phrase for /home/kali/private.key:  
kali@kali:~$ cat /home/kali/dtest.txt  
This is a test txt file for RSA  
kali@kali:~$ cat /home/kali/test.txt  
This is a test txt file for RSA  
kali@kali:~$
```

OpenSSL is a powerful secure socket layer cryptographic library, including the main cryptographic algorithms, commonly used key and certificate package management functions and SSL protocol, and provides a wealth of applications for testing or other purposes.

1. Symmetric encryption - The standard command used for symmetric encryption is `openssl enc`
2. One-way encryption – The standard command to use for one-way encryption is `openssl dgst`
3. Generate password – The standard command to generate a password is `openssl password`
4. Generate key pair – First, you need to use the `openssl genrsa` standard command to generate the private key and use the `rsa` standard command to extract the public key from the private key.
5. Generate random numbers – The standard command used to generate random numbers is `openssl rand`.

Security Monitoring and Vulnerability

The function of Nagios is to monitor services and hosts, but it does not include these functions itself. All monitoring and detection functions are completed through various plug-ins. After Nagios is started, it will automatically call the plug-in periodically to check the server status. At the same time, Nagios will maintain a queue, and all the status information returned by the plug-in will enter the queue. Display the status results through the web.

There are some tips can be used to increase security in a network. First is do not run Nagios as root. There needs to be normal user named nagios. If Nagios runs as root, when Nagios is attacked, the attacker can do whatever he wants with the user's system. And then project remote agents. Remote agents include NRPE, NSClient, SNMP, etc.

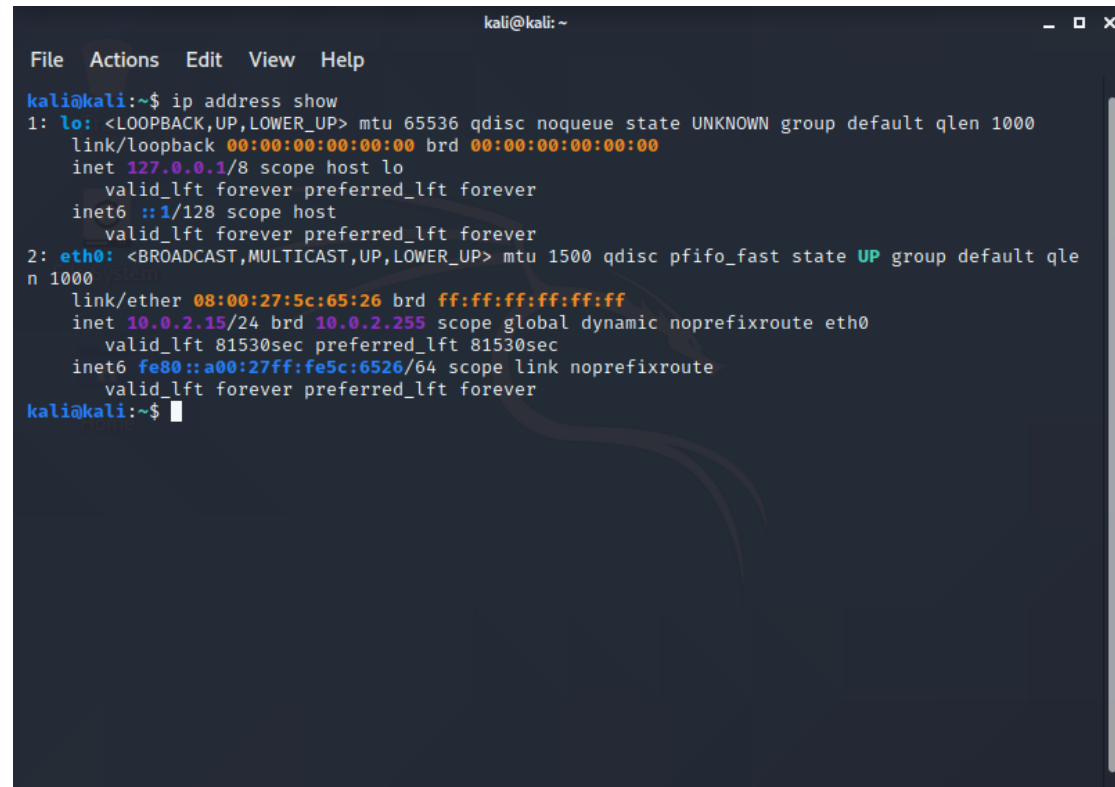
1. Advanced Persistent Threats (APTs) – usually lurking in the software for a long time to obtain huge goals, a variety of custom software may become their tools.
2. Macro and Scripting Viruses – it will be attached to the document and executed and propagated using the macro programming function of the document application
3. Backdoor (trapdoor) – Any mechanism that bypasses normal security checks, it may allow unauthorized access to programs or functions of infected systems.
4. Downloaders – Code that installs other items on a machine that is under attack. It is normally included in the malware code first inserted on to a compromised system to then import a larger malware package.
5. Logic bombs – It is malicious code embedded in normal software and executed under certain circumstances. These specific situations include changing files, special program input sequences, specific times, or dates, etc. Malicious code may delete files, crash the host computer, or cause other damage.
6. Rootkit – It is a special kind of malware. Its function is to hide itself and specified files, processes, network links and other information on the installation target. It is more common to see that rootkits are generally used in combination with Trojans, backdoors, and other malicious programs.
7. Spyware – It is software that installs backdoors on users' computers and collects user information without the user's knowledge. It can weaken users' material control over their experience, privacy, and system security. Use users' system resources, including programs installed on their computer or collect, use, and distribute users' personal or sensitive information
8. Trojan horse – It is an unauthorized remote-control program hosted in the computer. It is enough to open system permissions, leak user information, and even steal the entire computer management permission without the knowledge of the computer administrator.
9. Worm – It is a common computer virus. It is an independent program that

can run without the intervention of a computer user. It spreads by continuously obtaining some or all control rights on a vulnerable computer in the network.

10. Zombie bot – Zombie programs infect the vulnerable computers through chat rooms and file sharing networks. The information saved by these infected computers can be accessed by hackers

Kail Linux

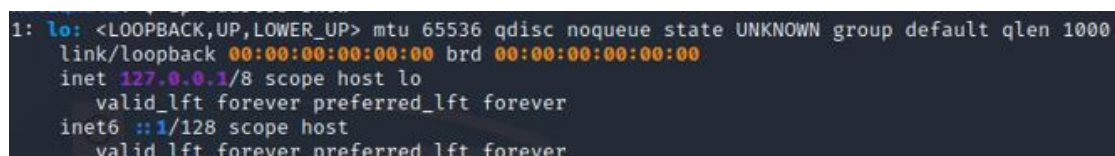
1. What's your computer's IP address for its current Internet connection? (How can you tell the difference between your Ethernet IP and your wireless IP if you have both connections active?)



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ ip address show  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 08:00:27:5c:65:26 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0  
        valid_lft 81530sec preferred_lft 81530sec  
    inet6 fe80::a00:27ff:fe5c:6526/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
kali@kali:~$
```

2. How can you determine the IP address associated with a given host name?

Type “ip address show” it will list all the connection and you can see the host name followed by the IP address.



```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever
```

3. How can you determine the host name(s) associated with a given IP address?

Type “ip address show” it will list all the connection and you can see the host name followed by the IP address. `inet 127.0.0.1/8 scope host lo`

4. How can you copy a file from one computer to another? Or more to the point, if you create a file on the Kali virtual machine and you want to put it someplace where you can save it, how do you go about it from the Kali command-line interface?

Type “scp [Local Linux system file path] [remote username@IP address:/Directory]”.

E.g. scp user/home/local/* [root@127.0.0.1:/user/home/local](#)

After you enter this command, system will ask you to enter the password of root server then start to copy data remotely.

5. How can you tell whether there's a process listening on a given port (e.g. port 80 or port 22) on a given host?

Type “netstat -an|grep 80” or “netstat -an|grep 22”

E.g. Type “netstat -an|grep 80”

```
kali@kali:~$ netstat -an|grep 80
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
unix 2      [  ]        DGRAM          1806         -
        /run/systemd/jo
urnal/syslog
unix 2      [ ACC ]     STREAM        LISTENING     20880        -
        /tmp/ssh-bByRnd
b4Yuzp/agent.795
unix 2      [ ACC ]     STREAM        LISTENING     20380        795/xfce4-session
        @/tmp/.ICE-unix
/795
unix 3      [  ]        STREAM        CONNECTED     20347        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     23357        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     20308        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     21277        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     22661        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     22473        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     27715        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     22701        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     22672        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     21256        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     21127        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     23011        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     22045        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     21061        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     23352        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     23608        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     23017        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     23002        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     23609        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     21280        -
        /run/user/1000/
unix 3      [  ]        STREAM        CONNECTED     21383        803/dbus-daemon
        /run/user/1000/
bus
unix 3      [  ]        STREAM        CONNECTED     22051        803/dbus-daemon
        /run/user/1000/
bus
unix 2      [  ]        DGRAM          20309        803/dbus-daemon
unix 3      [  ]        STREAM        CONNECTED     18008        -
```

6. How can you tell which ports have processes listening on them on a given host?

Type “netstat -anulp”

E.g.

```
kali@kali:~$ netstat -anulp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program nam
e
udp        0      0 10.0.2.15:68            10.0.2.2:67            ESTABLISHED -
```

7. How can you retrieve and save a given web page (say <http://google.com/> in a file on your system?

Type “wget <http://google.com/>”

```
kali@kali:~$ wget http://google.com/
--2020-08-31 21:39:56-- http://google.com/
Resolving google.com (google.com)... 74.125.193.113, 74.125.193.138, 74.125.193.101, ...
Connecting to google.com (google.com)|74.125.193.113|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2020-08-31 21:39:56-- http://www.google.com/
Resolving www.google.com (www.google.com)... 74.125.193.103, 74.125.193.104, 74.125.193.147, ..
Connecting to www.google.com (www.google.com)|74.125.193.103|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html      [  =>  ] 12.15K --KB/s  in 0s
2020-08-31 21:39:56 (212 MB/s) - 'index.html' saved [12442]

kali@kali:~$
```

8. How can you view the HTTP headers sent back from a specified web server when you request one of its pages?

Type `curl -I "http://google.com/"`

A terminal window titled 'kali@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The terminal shows the command 'curl -I "http://google.com/"' and its output: 'HTTP/1.1 301 Moved Permanently', 'Location: http://www.google.com/', 'Content-Type: text/html; charset=UTF-8', 'Date: Tue, 01 Sep 2020 01:40:26 GMT', 'Expires: Thu, 01 Oct 2020 01:40:26 GMT', 'Cache-Control: public, max-age=2592000', 'Server: gws', 'Content-Length: 219', 'X-XSS-Protection: 0', and 'X-Frame-Options: SAMEORIGIN'. The prompt 'kali@kali:~\$' is visible at the bottom.

```
kali@kali:~$ curl -I "http://google.com/"
HTTP/1.1 301 Moved Permanently
Location: http://www.google.com/
Content-Type: text/html; charset=UTF-8
Date: Tue, 01 Sep 2020 01:40:26 GMT
Expires: Thu, 01 Oct 2020 01:40:26 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 219
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN

kali@kali:~$
```

9. [Super bonus question] Is there a command-line-only way to view the HTTP headers that *my* computer sends when I run the commands in the previous two questions?

We can use `wget` and `curl` command to check the header.

10. What is the difference between Kali Linux and other Linux distributions? How does it support computer security?

The publisher is different, and the built-in software is quite different. The design purposes of them are different. Kali is designed for security audits and is a free system, so the software that comes with them is different. Kali Linux has many things in it, like any other flavor of Linux it uses the Linux kernel. Kali Linux is a Penetration Testing Distribution. This means it is set up to break into things by default (skill of user dependent). The different between Kali Linux and another Linux is: Kali is a Desktop environment; another Linux is an Operating System kernel.

Kali Linux is equipped with hundreds of useful security tools belonging to different categories, such as vulnerability analysis, wireless attacks, web applications, development tools stress testing and forensics tools.

Discussion

Based on using cryptography can achieve the following purposes:

1. Authentication – This process to prove the identity of an entity can be based on something you know, such as a password; something you have, such as an encryption key.

2. Data confidentiality – This property makes information is not made available or disclosed to unauthorized individuals, entities, or processes.

3. Data integrity – This property refers to data that has not been changed, destroyed, or lost in an unauthorized or accidental manner. The need for data integrity is especially evident if data is transmitted across a nonsecure network, such as the internet, where a man-in-the-middle attack can easily be mounted. Integrity is enforced by mathematical functions applied to the message being transmitted.

4. Nonrepudiation - Repudiation is the denial by one of the entities involved in a communication of having participated in all or part of the communication. Nonrepudiation is protection against repudiation and can be of two types.

- Nonrepudiation with proof of origin provides the recipient of data with evidence that proves the origin of the data and thus protects the recipient against an attempt by the originator to falsely deny sending the data. Its purpose is to prove that a particular transaction took place, by establishing accountability of information about a particular event or action to its originating entity.
- Nonrepudiation with proof of receipt provides the originator of data with evidence proving that data was received as addressed and thus protects the originator against an attempt by the recipient to falsely deny receiving the data.

In most cases, the term nonrepudiation is used as a synonym of nonrepudiation with proof of origin. Like integrity, nonrepudiation is based on mathematical functions applied to the data being generated during the transaction.

We can accomplish those goals using encryption tools to protect your personal data and privacy, etc. however, there still is some vulnerability that threaten the confidentiality, integrity, availability, and access control of the system or its application data. Using security monitoring to monitor network and host activities in real time, monitor and analyze user and system behavior, audit system configuration and vulnerabilities, assess the integrity of sensitive systems and data, identify attack behaviors, perform statistics and tracking of abnormal behaviors, identify security violations Regulatory actions, using decoy servers to

record hacking behaviors and other functions, enable administrators to effectively monitor, control, and evaluate networks or host systems.

The advantages of Kali Linux: integration of various tools. According to the data I collected, Kali Linux is not for beginner users, but a cool “hacking operating system”. Kali Linux is a collection of excellent security tools for professional users, it has incredible benefits, but for some users with bad intentions, it can also cause a lot of trouble. During the report, I found Kali Linux is more difficult to get started with than other Linux systems, but the system itself is an excellent system with high confidentiality.