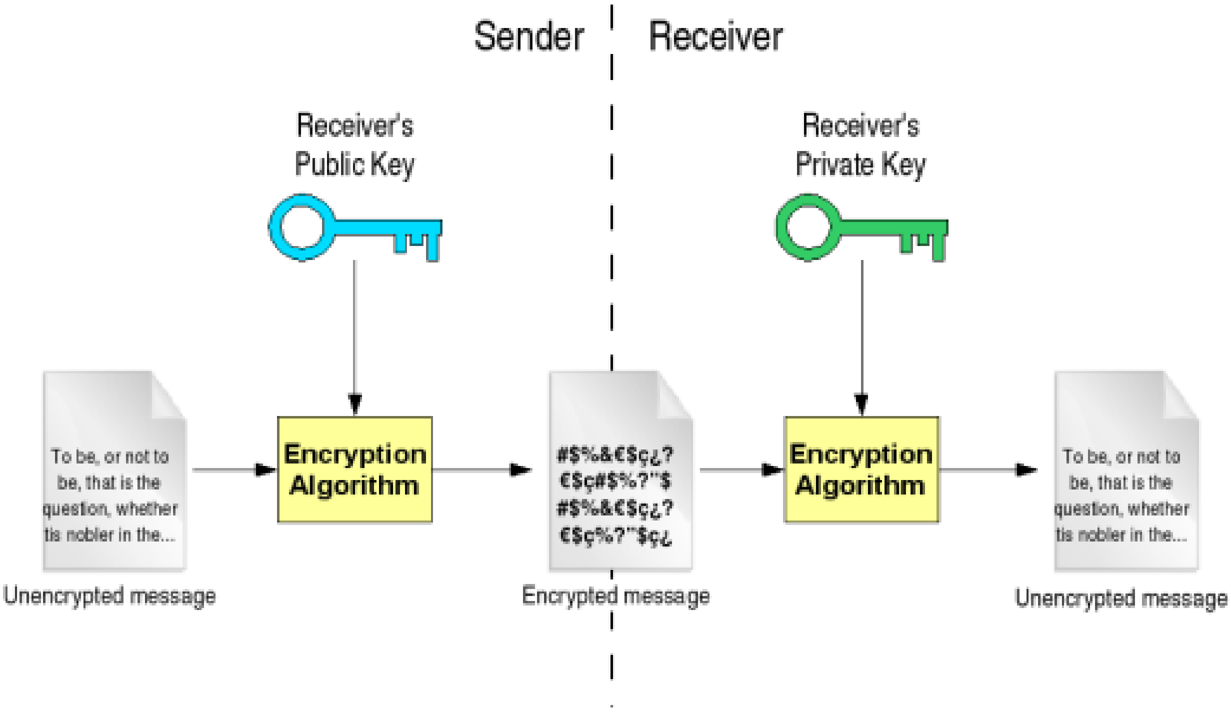
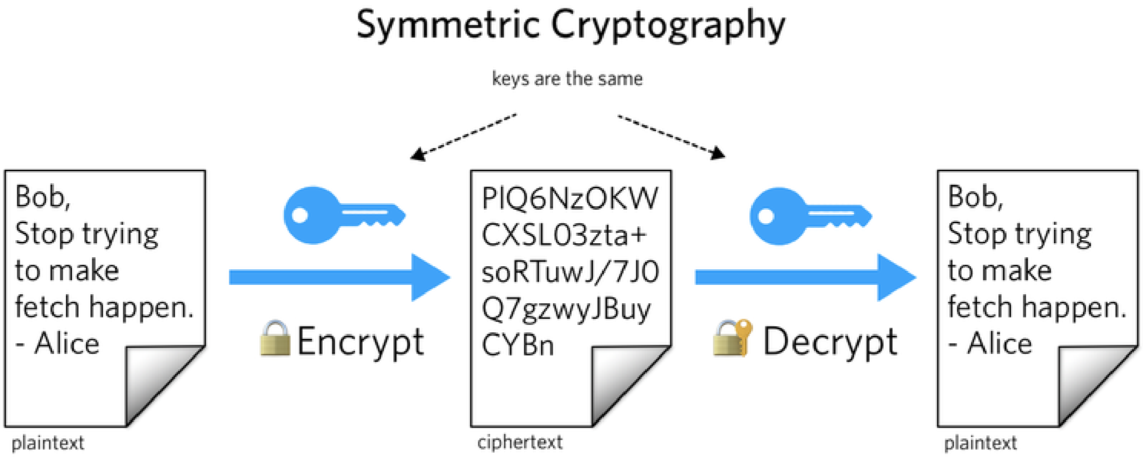
Cryptography in Linux

- From TLS to authentication, “crypto” is used for a lot more than just currencies. Security should be part of every developer's toolkit and cryptography a fundamental building block for the libraries and tools we use to protect our data and applications. This post will dive into modern cryptography, an overview of how it works, and its everyday use cases — including how Twilio uses public-key crypto in our Authy application and to secure our API.



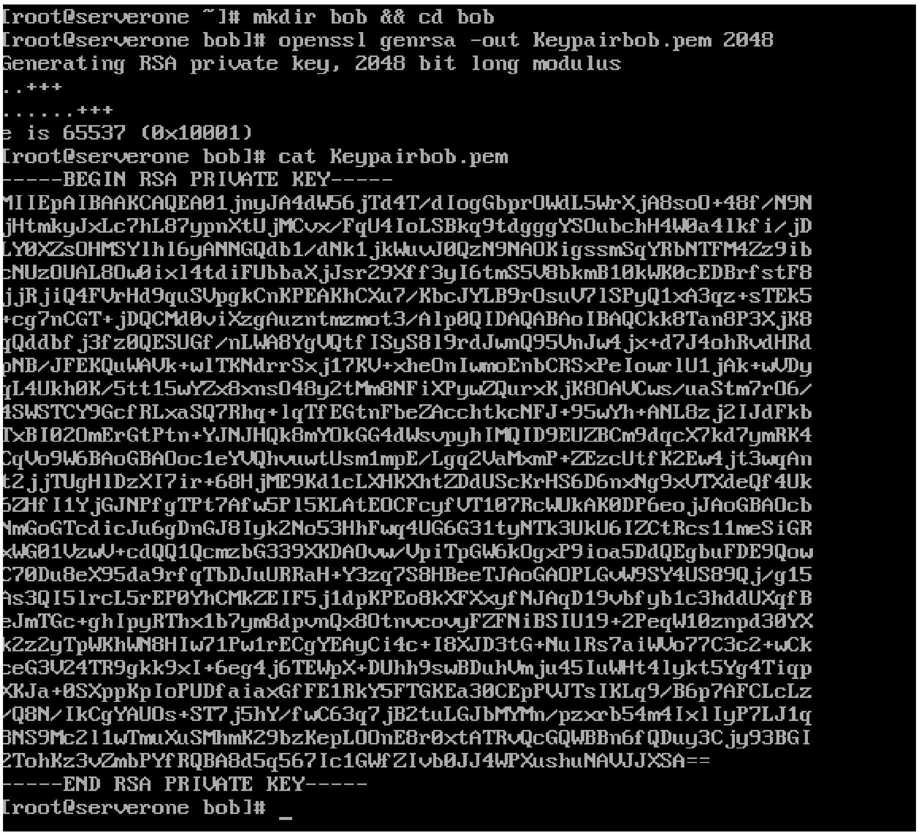


<u>Generating public key and private Key:</u>

bob user:

Use openssl to create private key and public key

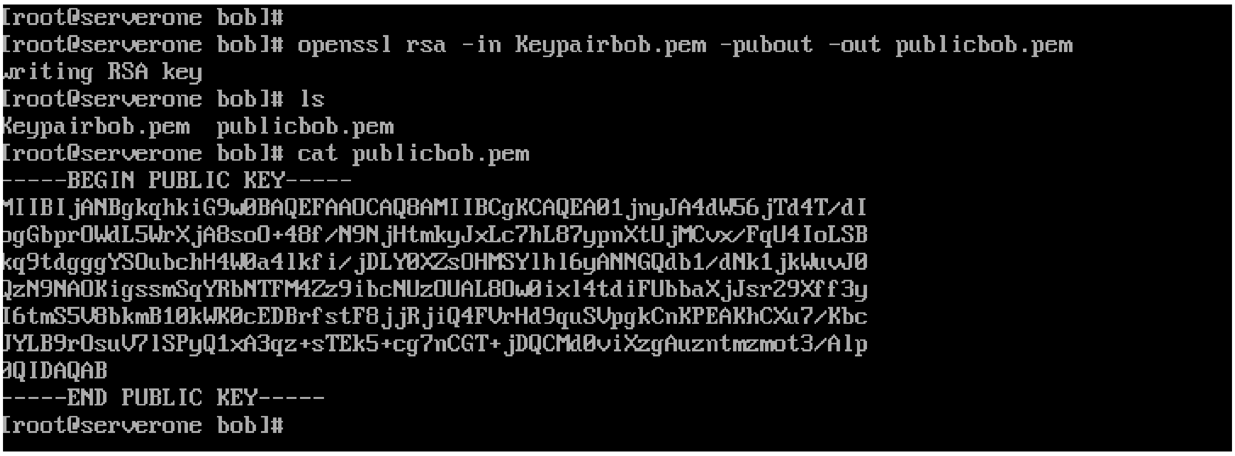
- # openssl genrsa –out Keypairbob.pem 2048



```
[root@serverone ~]# mkdir bob && cd bob
[root@serverone bob]# openssl genrsa -out Keypairbob.pem 2048
Generating RSA private key, 2048 bit long modulus
..+++
......+++
e is 65537 (0x10001)
[root@serverone bob]# cat Keypairbob.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA01jnyJA4dW56jTd4T/dIogGbprOWdL5WrXjA8soO+48f/N9N
jHtmkyJxLc7hL87ypnXtUjMCvx/FqU4IoLSBkq9tdgggYSOubchH4W0a4lkfi/jD
LY0XZsOHMSYlhl6yANNGQdb1/dNk1jkWuvJ0QzN9NAOKigssmSqYRbNTFM4Zz9ib
cNUzOUAL8Ow0ixl4tdiFUbbaXjJsr29Xff3yI6tmS5V8bkmB10kWK0cEDBrfstF8
jjRjiQ4FVrHd9quSVpgkCnKPEAKhCXu7/KbcJYLB9rOsuV7lSPyQ1xA3qz+sTEk5
+cg7nCGT+jDQCMd0viXzgAuzntmzmot3/Alp0QIDAQABAoIBAQCkk8Tan8P3XjK8
yQddbfj3fz0QESUGf/nLWA8YgVQtfISyS8l9rdJwmQ95VnJw4jx+d7J4ohRvdHRd
pNB/JFEKQuWAVk+wlTKNdrrSxj17KV+xheOnIwmoEnbCRSxPeIowrlU1jAk+wVDy
qL4Ukh0K/5tt15wYZx8xnsO48y2tMm8NFiXPywZQurxKjK8OAVCws/uaStm7rO6/
4SWSTCY9GcfRLxaSQ7Rhq+lqTfEGtnFbeZAcchtkcNFJ+95wYh+ANL8zj2IJdFkb
TxBI02OmErGtPtn+YJNJHQk8mYOkGG4dWsvpyhIMQID9EUZBCm9dqcX7kd7ymRK4
CqVo9W6BAoGBAOoc1eYVQhvuwtUsm1mpE/Lgq2VaMxmP+ZEzcUtfKZEw4jt3wqAn
t2jjTUgHlDzXI7ir+68HjME9Kd1cLXHKXhtZDdUScKrHS6D6nxNg9xVTXdeQf4Uk
5ZHfI1YjGJNPfgTPt7Afw5Pl5KLAtEOCFcyfVT107RcWUkAK0DP6eojJAoGBAOcb
NmGoGTcdicJu6gDnGJ8Iyk2No53HhFwq4UG6G31tyNTk3UkU6IZCtRcs11meSiGR
xWG01VzwV+cdQQ1QcmzbG339XKDAOvw/VpiTpGW6kOgxP9ioa5DdQEgbuFDE9Qow
C70Du8eX95da9rfqTbDJuURRaH+Y3zq7S8HBeeTJAoGAOPLGvW9SY4US89Qj/g15
As3QI5lrcL5rEP0YhCMkZEIF5j1dpKPEo8kXFXxyfNJAqD19vbfyb1c3hddUXqfB
eJmTGc+ghIpyRThx1b7ym8dpvnQx8OtnvcovyFZFNiBSIU19+2PeqW10znpd30YX
k2z2yTpWKhWN8HIw71Pw1rECgYEAyCi4c+I8XJD3tG+NulRs7aiWVo77C3c2+wCk
ceG3VZ4TR9gkk9xI+6eg4j6TEWpX+DUhh9swBDuhVmju45IuWHt4lykt5Yg4Tiqp
XKJa+0SXppKpIoPUDfaiaxGfFE1RkY5FTGKEa30CEpPVJTsIKLq9/B6p7AFCLcLz
/Q8N/IkCgYAUOs+ST7j5hY/fwC63q7jB2tuLGJbMYMn/pzxrb54m4IxlIyP7LJ1q
3NS9Mc2l1wTmuXuSMhmK29bzKepLOOnE8r0xtATRvQcGQWBBn6fQDuy3Cjy93BGI
ZTohKz3vZmbPYfRQBA8d5q567Ic1GWfZIvb0JJ4WPXushuNAVJJXSA==
-----END RSA PRIVATE KEY-----
[root@serverone bob]# _
```

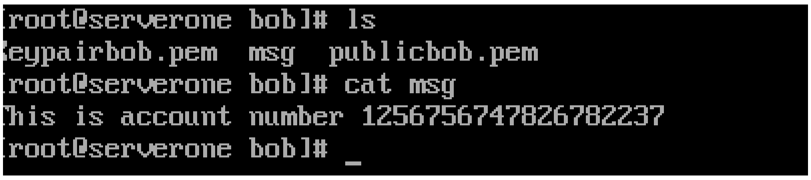
Generating the public Key:

- Openssl rsa –in Keypairbob.pem –pubout –out publicbob.pem



```
[root@serverone bob]#
[root@serverone bob]# openssl rsa -in Keypairbob.pem -pubout -out publicbob.pem
writing RSA key
[root@serverone bob]# ls
Keypairbob.pem  publicbob.pem
[root@serverone bob]# cat publicbob.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA01jnyJA4dW56jTd4T/dI
pgGbprOWdL5WrXjA8soO+48f/N9NjHtmkyJxLc7hL87ypnXtUjMCvx/FqU4IoLSB
kq9tdgggYSOubchH4W0a4lkfi/jDLY0XZsOHMSYlhl6yANNGQdb1/dNk1jkWuwJ0
QzN9NAOKigssmSqYRbNTFM4Zz9ibcNUzOUAL8Ow0ixl4tdiFUbbaXjJsr29Xff3y
I6tmS5V8bkmB10kWK0cEDBrfstF8jjRjiQ4FVrHd9quSVpgkCnKPEAKhCXu7/Kbc
JYLB9rOsuV7lSPyQ1xA3qz+sTEk5+cg7nCGT+jDQCMd0viXzgAuzntmzmot3/Alp
0QIDAQAB
-----END PUBLIC KEY-----
[root@serverone bob]#
```

- Create the file msg encrypt the msg:

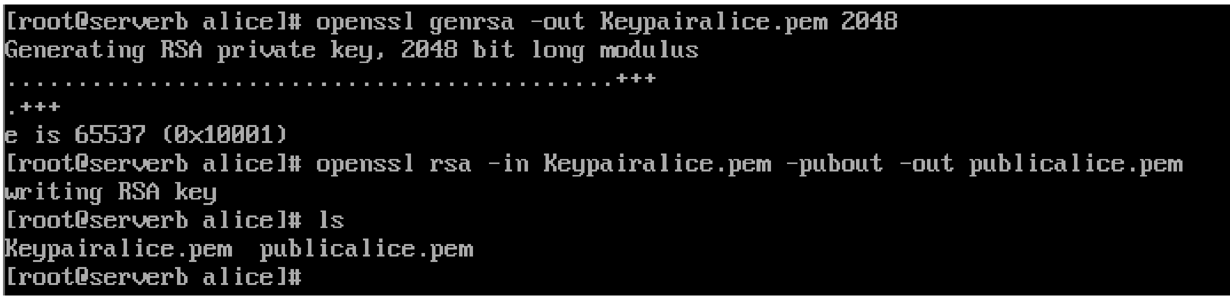


```
[root@serverone bob]# ls
Keypairbob.pem  msg  publicbob.pem
[root@serverone bob]# cat msg
This is account number 1256756747826782237
[root@serverone bob]# _
```

Generating the private and public Key:

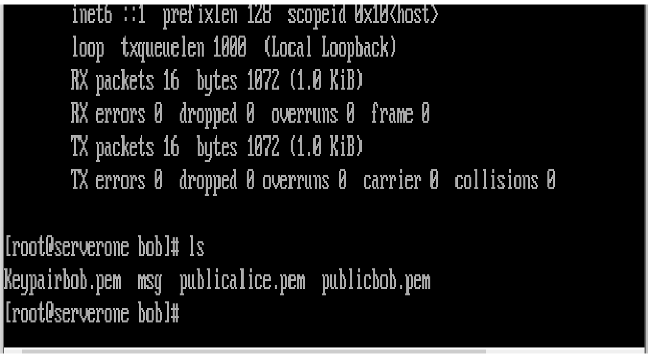
alice user:

- Steps are same as bob:



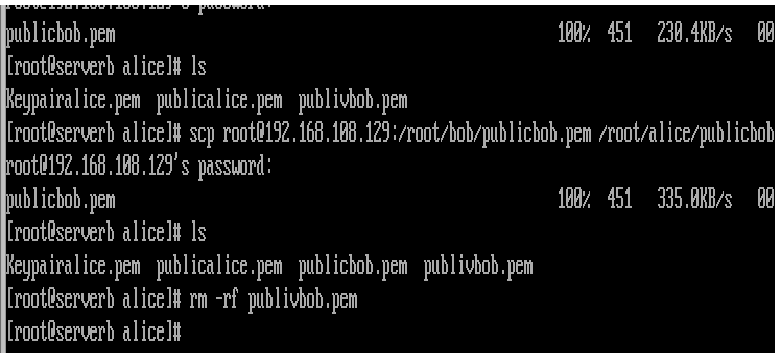
```
[root@serverb alice]# openssl genrsa -out Keypairalice.pem 2048
Generating RSA private key, 2048 bit long modulus
...........................................+++
.+++
e is 65537 (0x10001)
[root@serverb alice]# openssl rsa -in Keypairalice.pem -pubout -out publicalice.pem
writing RSA key
[root@serverb alice]# ls
Keypairalice.pem  publicalice.pem
[root@serverb alice]#
```

Share the public key between alice and bob:



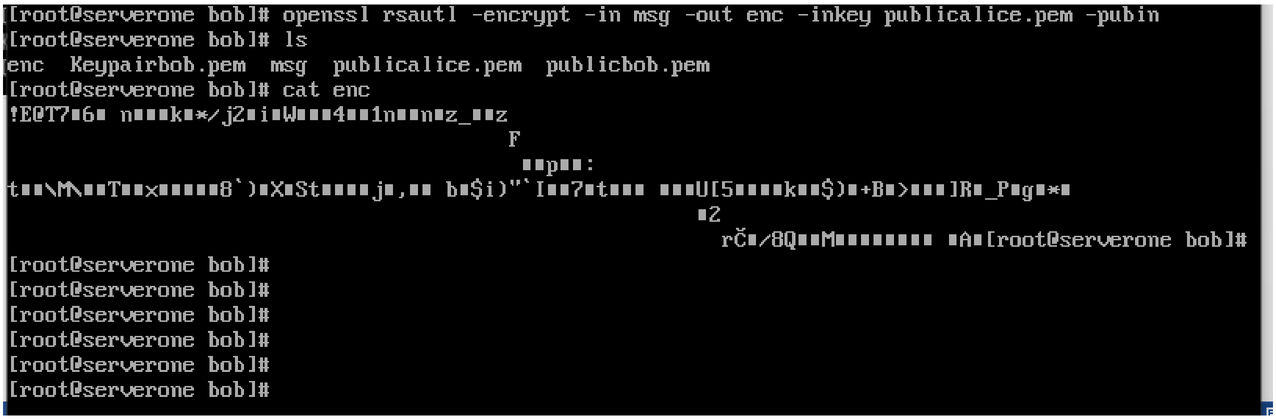
```
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 1072 (1.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1072 (1.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@serverone bob]# ls
Keypairbob.pem  msg  publicalice.pem  publicbob.pem
[root@serverone bob]#
```



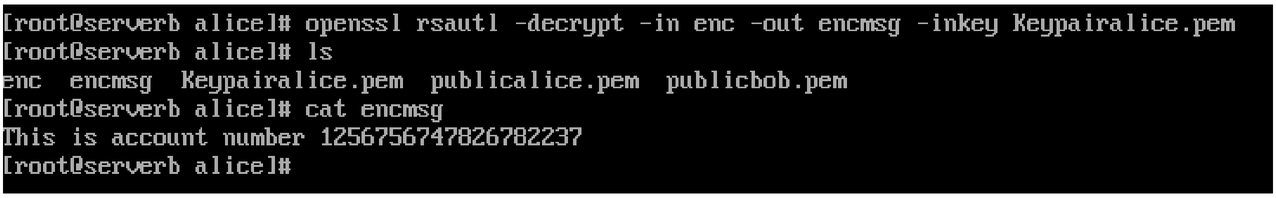
```
root@192.168.108.129's password:
publicbob.pem                                         100%  451   230.4KB/s   00
[root@serverb alice]# ls
Keypairalice.pem  publicalice.pem  publivbob.pem
[root@serverb alice]# scp root@192.168.108.129:/root/bob/publicbob.pem /root/alice/publicbob
root@192.168.108.129's password:
publicbob.pem                                         100%  451   335.0KB/s   00
[root@serverb alice]# ls
Keypairalice.pem  publicalice.pem  publicbob.pem  publivbob.pem
[root@serverb alice]# rm -rf publivbob.pem
[root@serverb alice]#
```

- ➢ Encrypt the msg in bob by the public key of alice
- ➢ # openssl rsault –encrypt –in msg -out enc -inKey publicalice.pem pubin



```
[root@serverone bob]# openssl rsautl -encrypt -in msg -out enc -inkey publicalice.pem -pubin
[root@serverone bob]# ls
enc  Keypairbob.pem  msg  publicalice.pem  publicbob.pem
[root@serverone bob]# cat enc
!E@T7▪6▪ n▪▪▪k▪*/j2▪i▪W▪▪▪4▪▪1n▪▪n▪z_▪▪z
                                           F
                                             ▪▪p▪▪:
t▪▪\M\▪▪T▪▪x▪▪▪▪▪8`)▪X▪St▪▪▪▪j▪,▪▪ b▪$i)"`I▪▪7▪t▪▪▪ ▪▪▪U[5▪▪▪▪k▪▪$)▪+B▪>▪▪▪]R▪_P▪g▪*▪
                                                             ▪2
                                                               rČ▪/8Q▪▪M▪▪▪▪▪▪▪▪ ▪A▪[root@serverone bob]#
[root@serverone bob]#
[root@serverone bob]#
[root@serverone bob]#
[root@serverone bob]#
[root@serverone bob]#
[root@serverone bob]#
```

- ➢ Share the enc file to alice decrypt by private key of alice:
- ➢ Openssl result –decrypt –in enc –out encmsg –inkey Keypairalice.pem



```
[root@serverb alice]# openssl rsautl -decrypt -in enc -out encmsg -inkey Keypairalice.pem
[root@serverb alice]# ls
enc  encmsg  Keypairalice.pem  publicalice.pem  publicbob.pem
[root@serverb alice]# cat encmsg
This is account number 1256756747826782237
[root@serverb alice]#
```