

Cryptographie et sécurité  
**IFT-606**

Devoir 1 - Cryptographie et attaques

Amandine Fouillet - 14 130 638  
Frank Chassing - 14 153 710

19 février 2015



## Table des matières

<b>1</b>	<b>Wi-Fi</b>	<b>4</b>
1.1	Fonctionnement des trois algorithmes de chiffrement . . . . .	4
1.2	Points faibles et attaques possibles des trois algorithmes de chiffrement . . . . .	4
1.3	Aircrack-ng . . . . .	4
<b>2</b>	<b>Chiffrement et signature</b>	<b>5</b>
2.1	Génération d'une paire de clé RSA . . . . .	5
2.2	Création d'un fichier contenant la partie publique de la clé RSA . . . . .	5
2.3	Chiffrement de la partie privée générée . . . . .	5
2.4	Chiffrement d'un message . . . . .	6
2.5	Déchiffrement d'un message . . . . .	6
2.6	Signature du fichier . . . . .	7
<b>3</b>	<b>Attaque décortiquée</b>	<b>7</b>

## Table des figures

1	Génération de la paire . . . . .	5
2	Fichier obtenu . . . . .	5
3	Exécution de la commande . . . . .	5
4	Clé publique . . . . .	5
5	Exécution de la commande . . . . .	6
6	Fichier cle.pem . . . . .	6
7	Fichier message.txt . . . . .	6
8	Exécution de la commande . . . . .	6
9	Fichier messageC.txt . . . . .	7
10	Exécution de la commande . . . . .	7
11	Fichier messageD.txt . . . . .	7
12	Exécution de la commande . . . . .	7
13	Fichier fic.sig . . . . .	7
14	Vérification de la signature . . . . .	8

# 1 Wi-Fi

## 1.1 Fonctionnement des trois algorithmes de chiffrement

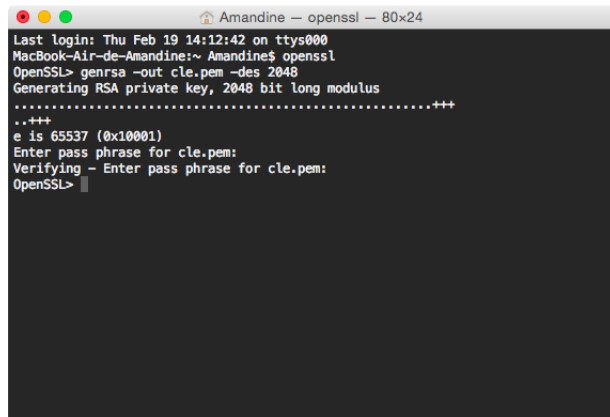
## 1.2 Points faibles et attaques possibles des trois algorithmes de chiffrement

## 1.3 Aircrack-ng

## 2 Chiffrement et signature

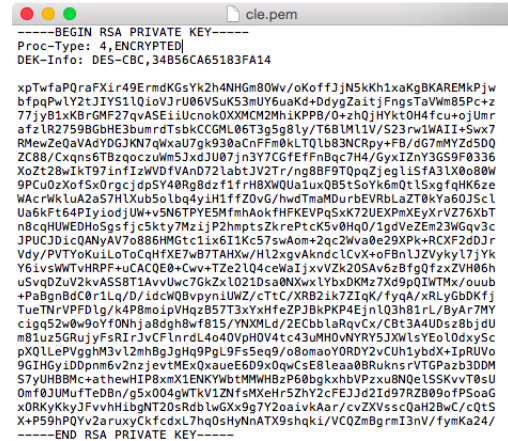
### 2.1 Génération d'une paire de clé RSA

Pour générer une paire de clé RSA d'une taille de 2048 bits protégée par un mot de passe, on exécute la commande suivante : `genrsa -out cle.pem -des 2048` (1). Le fichier généré `cle.pem` (FIGURE 2) contient maintenant la paire de clé RSA d'une taille de 2048.



```
Amandine — openssl — 80x24
Last login: Thu Feb 19 14:12:42 on ttys000
MacBook-Air-de-Amandine:~ Amandine$ openssl
OpenSSL> genrsa -out cle.pem -des 2048
Generating RSA private key, 2048 bit long modulus
.....+++
..+++
e is 65537 (0x10001)
Enter pass phrase for cle.pem:
Verifying - Enter pass phrase for cle.pem:
OpenSSL>
```

FIGURE 1 – Génération de la paire



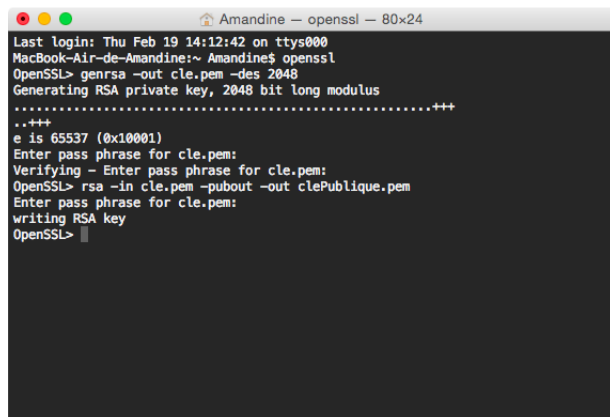
```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-CBC,34B56CA651B3FA14

xpTwaPQraFX1r49ErmDKGsYk2h4NHGm80Wv/oKoffJjN5kKh1xaKgBKAREMkPjw
bfpqPwLY2tJiY51LQioVjrU06VsuK53mUY6uaKd+DdygZaitjFngsTaWmB5Pc+z
77jy81xKBrGMF27qvASE1iUcnok0XMXCM2Mh1KPPB/B+zhQjHYkt0H4fCu+oJUmR
afz1R2759BqbHE3bumrdTsbCCML06T3gSg8ly/76bUW11V/S23rvWAI1-Swx7
RMewZeQaVAdYDGJKN7qWxaU7gk930acnFFm0kLT01b83NCRpy+FB/d67mMYZd5DQ
ZC88/Cxqns6T8zqoczUwm5JxdJU07jn3Y7CGfEfFnBgc7H4/GyxIZnY3G59F0336
XoZt28wIkT97lnfIzWVDfVAnd72LabtJV2Tr/ng8BF9TQpqZegLisfa3LX0o80W
9PCu0zXofSx0rgcjdpSY40Rg8dzf1frH8XWQUa1uxQ85tSoYk6mQtLSxgfqHK6ze
WAcrrWkLuA2a57H1Xub5o1bq4y1H1ffZ0vG/hwdTmaDurbEVRbLaZT0kYa60J5cL
Ua6Kf64PIyiodjUW+vS6TPYE5MfmaAokfHPKEVPq5xK72UEXPMExYrV276xBT
n8cqUHEdHo5gofjCskty7Mz1jP2hmptsZkrePck50Hq0/1gdv6ZEn23W6qV3C
JPUCJd1c0AMyAV7o886HMcT11x611Kc57swom+Zqc2Wva0e29PK+RCXF2d0Jr
Vdy/PVTYoKuiloToCqHfXE7wB7TAHXw/H12xgvaKndc1CvX+oF8n1JZVkyk17jYk
Y6ivsWNTvHRFP+uCACQEO+Cwv+TZe2LQ4ceWaIjxvVZK20SAv6z8fg0fzxZVH06h
uSvQDzuV2kvaS58T1AvvUwc7GkZxL021Dsa0NXw1YbxDKMz7Xd9pQIWTMx/ouub
+PaBgnBdC0r1Lq/D/ldcWQBvpyinUWZ/cTtC/XRB2ik7ZiQk/fyqA/xRLyGbDKfj
TueTnrvPFDlg/k4P8moipVHzB57T3xYxHfZPJBkPKP4EjnlQ3h81rL/ByAr7MY
c1gq52w0w90yFDNhja8dgh8wfb15/YNXMLd/2ECbbLaRqvCx/CBT3A4U0sz8bjdU
n81uz56RuYjFsf1JvCFlnrdL4o40VpHOVatc43uMH0uVYR5JXwLsYeo10dxy5c
pX01LePVgghM3v12mhBgJgHq9Pgl9fs5eq9/o8omaoYORDY2vCUh1ybdX+IprUvo
9GIHGyIDdpnm6v2nzjevtMEXQxaveE6D9xQdwCsE8lea0BRuknsrvTGPazb3DDM
S7yUHBMc+athewHIP8xmX1ENKYbttMMWHBzP6bgkxhbVPZxu8N0eLSKvT0sU
0mf0JUMufTeDn/g5x004gWTKv1ZNFsMXeHr5ZhY2cFEJ3d2iD97RZB09oFpSoaG
x0RKyKkyJFvvhHigNT20sRdbLwGx9g7Y2oaivkaAr/cvZXVsscQaH2BwC/c0tS
X+P59hPQYvZaruxyCkcdxL7hq0sHyNnATX9shqk1/VcQZmBgRmI3v/fymKa24/
-----END RSA PRIVATE KEY-----
```

FIGURE 2 – Fichier obtenu

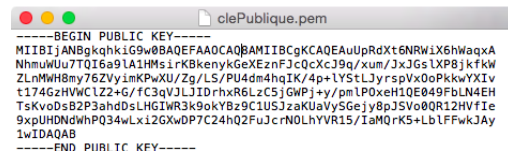
### 2.2 Création d'un fichier contenant la partie publique de la clé RSA

Pour créer un fichier contenant seulement la partie publique de la clé RSA on exécute la commande suivante : `rsa -in cle.pem -pubout -out clePublique.pem` (FIGURE 3). Le fichier généré `clePublique.pem` (FIGURE 4) contient maintenant la clé publique.



```
Amandine — openssl — 80x24
Last login: Thu Feb 19 14:12:42 on ttys000
MacBook-Air-de-Amandine:~ Amandine$ openssl
OpenSSL> genrsa -out cle.pem -des 2048
Generating RSA private key, 2048 bit long modulus
.....+++
..+++
e is 65537 (0x10001)
Enter pass phrase for cle.pem:
Verifying - Enter pass phrase for cle.pem:
OpenSSL> rsa -in cle.pem -pubout -out clePublique.pem
Enter pass phrase for cle.pem:
writing RSA key
OpenSSL>
```

FIGURE 3 – Exécution de la commande



```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAU0pRdXt6NRW1X6HwaqXA
NhmuUu7TQ16a91A1HMsirKBkenyKGeXEznFJcQcXc39q/xum/3xJGsLXP8jKfKw
ZLnMWH8my76ZVymKPwXU/Zg/L5/PU4dm4hqIK/4p+LYStLjyrspVx0oPKkwYXiv
t1746ZHVWCLZ2+g/fC3qVJLJIDrhxR6LzC5jGWPj+y/pmLP0xeH10E049FbLN4EH
TsKvo0sB2P3ahd0sLHG1WR3k9okY8z9CIUSJzakUaVysGejy8pJ5Vo0R12HVf1e
9xpUHDndwhPQ34wLx12GxwD7C24hQ2FujcrN0LhYVR15/IaM0rk5+Lb1FFwKJy
1wIDAQAB
-----END PUBLIC KEY-----
```

FIGURE 4 – Clé publique

### 2.3 Chiffrement de la partie privée générée

Pour chiffrer la partie privée générée, on exécute la commande suivante : `rsa -in cle.pem -des3 -out cle.pem` (FIGURE 5). Quand on réouvre le fichier `cle.pem` on remarque que le chiffrement a changé pour un chiffrement avec l'algorithme `des3` (FIGURE 6).

```

Amandine — openssl — 80x24
Last login: Thu Feb 19 14:12:42 on ttys000
MacBook-Air-de-Amandine:~ Amandine$ openssl
OpenSSL> genrsa -out cle.pem -des 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
e is 65537 (0x10001)
Enter pass phrase for cle.pem:
Verifying - Enter pass phrase for cle.pem:
OpenSSL> rsa -in cle.pem -pubout -out clePublique.pem
Enter pass phrase for cle.pem:
writing RSA key
OpenSSL> rsa -in cle.pem -des3 -out cle.pem
Enter pass phrase for cle.pem:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
OpenSSL>

```

FIGURE 5 – Exécution de la commande

```

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-ED3-CBC, AF82BA7AAB810D2F

BUYG+yj40In6CU+0Fgj5v8ur/1Xbf2zrA4rTmunie+lfv7R86Gdt75j0XBATEvtm
yp0lp2Bj0LSXXKyxAvwXL3/rYk0MngRudy062tQYd58jgB9Hf/RB0jKAgLtnPX
0ua8ANB4dwltDsYvP5Rwk1QqPvMRIez8HpaONR82D7YzDnW2vU8Tsvi/x9A9+4
emuq/mXX3BAjpbFkWLcwYv7yo0wy6xIGtccm0XqceEakCAveBuF8HE9jUVyngb5i
lWh8dAsN8dMg0BA0FavZ/Dfd3u7lwXV3dMS7GI1pf2PC7ymjfEiwDdQF12EkvV
AhJexN0/Pv/HZs0IFRhm03BK40ceNBRDSU0hSpG67PGXlMSV9FcfLOE040SAf4K
ecFe63i+hU0Yfm9r0KeNv7s0cjKUpLVGY9ESUIWA8ApA4FPis/sY0UujvGzKmvRZ
E5pThjJkvWv0LyAgNPCT+0z4mxY1SufxU4eR1PBKU0Foyh/6wI0Iwhg0MvPKG8T
jmsUiz0dHosbsV6gCn6Fdk0KZfMpk/ECLU1E0AAAXEk49tY3T/dTf4Za1s+uUK
ETP5tPy5P2KX5kFTnRAf2hGZrLstTba0ok7bNk+sf4/bbQR44nE668102qt9d0
7BFkTHfNn7tqnbPNyRxGdJJUhu/817e83oTwaTPZTupJxRTMd8k3AA3mh0815Zp
gbCsc1urzNt0Zp6PyRo1L/WdwBdkk8pA3831ulronsVuGhm108k5pbu7/PPIdDQ
xNNkeca90gz+07538YqcdC0zPp0IXwAMwetXsgnFEf6r864LD6k8Wf3sgYX0u1
ph2K0iSKx1vU8mLJ3wdLE0+o9r7a31jnxIdtdlnU/Csk3Txv01oR7KKxqTvwXLD5
hgXrRzz/kVb8wcEY24xsGdIrjgKmC4zJepeABJ0beE50e7CFhrYH83u2UNLPt7c6
IglAx1UmW+34oeIdKr/6adu3EEdMwHYt+/u/30YfC0RXPbIdnNP/ew0t1fA0V17
/KfBk0z0xKFX5xnVRf4bJfT80eAT8080sHaV6e+uJ2v+1YUSdCBVV1rcXy6d
xIdnJwc9jEwvr0FgWEL88+W/RwXkgRMe4UKX8vKzcZymN4cjoBv5/n+UaXkSd
2PgU094e60H0KHEVio2Hr7rC0u3gIEK3CJ13B8cbxftgBU7WdcCjo44lmGKx1Q
yTL17D2SvsdJJxyCrXzyaUxjv59Pd0AwrlVp5agPvKdb2cx86HwopG2oeFceU
YVLf9f1ykZG03vt03xpwG8APw3n5UURVuttyIxL9fciMm+5cpxyM9HH+4G2Abg7
SidpuRy0IkWDMp7WjPcu8pDbG0BxJM1Ys/8dNx8pIku856d1kLUIyIZ0wu0Vpxc
bxgXx12L26Gf880Mu2ukPsb0PovUek0bwuM6LWuAxs0T7LejWsp17e5j5JKB2Yv
yJyE3V5LlqVXB+219/knn0LkNbUUBGg9XAHxZVEFocTX09CgAdwZt2d8Rbo2xpd
jm6fLFLZ5lmfD9K5Zf3v6pJlUkH5j0Klvud/h1f7db10Rk5L0GbmKdFXRxt0t9P
-----END RSA PRIVATE KEY-----

```

FIGURE 6 – Fichier cle.pem

## 2.4 Chiffrement d'un message

Nous allons maintenant chiffrer le fichier message.txt (FIGURE 7) qui contient le message "OpenSSL is really cool!!!". Pour se faire, nous exécutons la commande suivante : `rsautl -encrypt -in message.txt -inkey cle.pem -out messageC.txt` (FIGURE 8). Le fichier messageC.txt contient le message crypté (FIGURE 9).

```

message
OpenSSL is really cool!!!

```

FIGURE 7 – Fichier message.txt

```

Amandine — openssl — 80x24
Last login: Thu Feb 19 14:12:42 on ttys000
MacBook-Air-de-Amandine:~ Amandine$ openssl
OpenSSL> genrsa -out cle.pem -des 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
e is 65537 (0x10001)
Enter pass phrase for cle.pem:
Verifying - Enter pass phrase for cle.pem:
OpenSSL> rsa -in cle.pem -pubout -out clePublique.pem
Enter pass phrase for cle.pem:
writing RSA key
OpenSSL> rsa -in cle.pem -des3 -out cle.pem
Enter pass phrase for cle.pem:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
OpenSSL> rsautl -encrypt -in message.txt -inkey cle.pem -out messageC.txt
Enter pass phrase for cle.pem:
OpenSSL>

```

FIGURE 8 – Exécution de la commande

## 2.5 Déchiffrement d'un message

Pour déchiffrer le message du fichier messageC.txt, on exécute la commande suivante : `rsautl -decrypt -in messageC.txt -inkey cle.pem -out messageD.txt` (FIGURE 10). On obtient le fichier messageD.txt qui contient le message déchiffré (FIGURE 11) qui correspond bien au message initial.

```

+ikΔ±][`7&#wK+òEQ>3x`ñC+/ñXN0-0,d
,ríAse$/$W&#:zC19=ÀðI8Xb |Mñj]fzòR2òU0Tt| æÃ""ÿ"·Ü0=B""·0"@oU
"BP(¼q="~CÈ9`xRi$W2ΔKòYyM"-òð\Næg·Δ3· 0l1-"èÜÿΔYiHD`SKByΔ"[11`$`m~
01RÿÜ·ÉÜ0C
0=Ej,71fEQqI8=46;
$012*#`cex16`$MI-c|

```

FIGURE 9 – Fichier messageC.txt

```

Amandine — openssl — 80x24
Last login: Thu Feb 19 14:12:42 on ttys000
MacBook-Air-de-Amandine:~ Amandine$ openssl
OpenSSL> genrsa -out cle.pem -des 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
..+++
e is 65537 (0x10001)
Enter pass phrase for cle.pem:
Verifying - Enter pass phrase for cle.pem:
OpenSSL> rsa -in cle.pem -pubout -out clePublique.pem
Enter pass phrase for cle.pem:
writing RSA key
OpenSSL> rsa -in cle.pem -des3 -out cle.pem
Enter pass phrase for cle.pem:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
OpenSSL> rsautl -encrypt -in message.txt -inkey cle.pem -out messageC.txt
Enter pass phrase for cle.pem:
OpenSSL> rsautl -decrypt -in messageC.txt -inkey cle.pem -out messageD.txt
Enter pass phrase for cle.pem:
OpenSSL>

```

FIGURE 10 – Exécution de la commande

```

messageD.txt
OpenSSL is really cool!!!

```

FIGURE 11 – Fichier messageD.txt

## 2.6 Signature du fichier

Pour signer le fichier, on exécute la commande suivante : *rsautl -sign -inkey cle.pem -in messageD.txt -out fic.sig* (FIGURE 12). La FIGURE 13 montre le fichier fic.sig obtenu.

```

Amandine — openssl — 80x24
Last login: Thu Feb 19 14:12:42 on ttys000
MacBook-Air-de-Amandine:~ Amandine$ openssl
OpenSSL> genrsa -out cle.pem -des 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
..+++
e is 65537 (0x10001)
Enter pass phrase for cle.pem:
Verifying - Enter pass phrase for cle.pem:
OpenSSL> rsa -in cle.pem -pubout -out clePublique.pem
Enter pass phrase for cle.pem:
writing RSA key
OpenSSL> rsa -in cle.pem -des3 -out cle.pem
Enter pass phrase for cle.pem:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
OpenSSL> rsautl -encrypt -in message.txt -inkey cle.pem -out messageC.txt
Enter pass phrase for cle.pem:
OpenSSL> rsautl -decrypt -in messageC.txt -inkey cle.pem -out messageD.txt
Enter pass phrase for cle.pem:
OpenSSL> rsautl -sign -inkey cle.pem -in messageD.txt -out fic.sig
Enter pass phrase for cle.pem:
OpenSSL>

```

FIGURE 12 – Exécution de la commande

```

fic.sig
^B&00%;v+>Ü „Ä„^Ü €1Âç@)„df=s-?p1M"/IF`Z9EQ)0oð-YÿWÜQH`
€'nt'òiln0'ΔI98Ü'ivgEUSÁi=çCÇgá(!„~#ðΔmE
SREcE
+0d$,ç=)0_æE2=ΔñE0ñi1 (,IÇawΔ) [_áv2±;i6àÜñYÁxçs-[
„Üvæ=MV"+kllIäI
ÜWI`"u=D-hZ}0iΔDÜÿIiDönNoh
6·}v 0'0`iæç$!..

```

FIGURE 13 – Fichier fic.sig

Pour vérifier la signature, on exécute la commande suivante : *rsautl -verify -pubin -inkey clePublique.pem -in fic.sig* (FIGURE 14). On obtient le résultat attendu.

```
MacBook-Air-de-Amandine:~ Amandine$ openssl
OpenSSL> genrsa -out cle.pem -des 2048
Generating RSA private key, 2048 bit long modulus
.....+++
..+++
e is 65537 (0x10001)
Enter pass phrase for cle.pem:
Verifying - Enter pass phrase for cle.pem:
OpenSSL> rsa -in cle.pem -pubout -out clePublique.pem
Enter pass phrase for cle.pem:
writing RSA key
OpenSSL> rsa -in cle.pem -des3 -out cle.pem
Enter pass phrase for cle.pem:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
OpenSSL> rsautl -encrypt -in message.txt -inkey cle.pem -out messageC.txt
Enter pass phrase for cle.pem:
OpenSSL> rsautl -decrypt -in messageC.txt -inkey cle.pem -out messageD.txt
Enter pass phrase for cle.pem:
OpenSSL> rsautl -sign -inkey cle.pem -in messageD.txt -out fic.sig
Enter pass phrase for cle.pem:
OpenSSL> rsautl -verify -pubin -inkey clePublique.pem -in fic.sig
OpenSSL is really cool!!!OpenSSL>
```

FIGURE 14 – Vérification de la signature

### 3 Attaque décortiquée