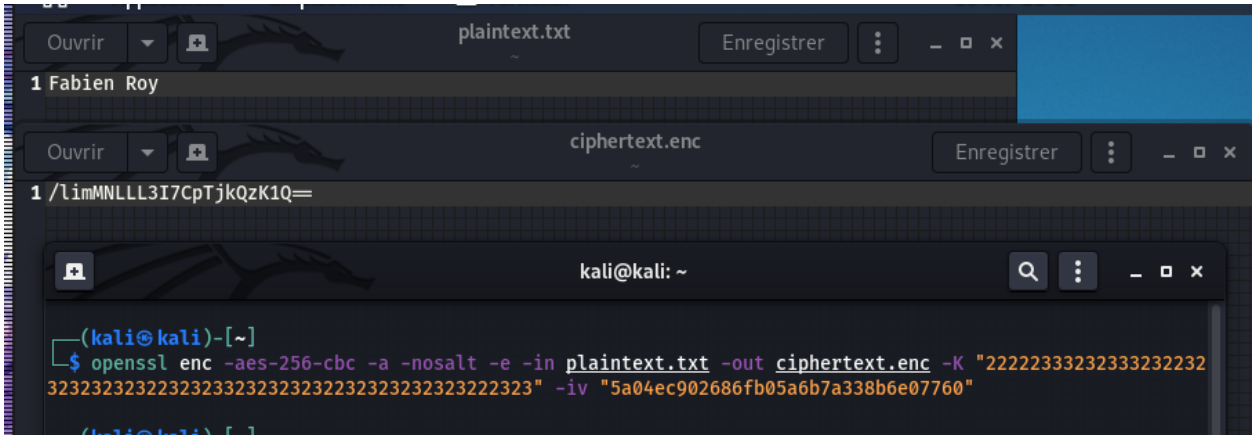
Fabien Roy
Automne 2021

TP1

Exercice 4

Question 1

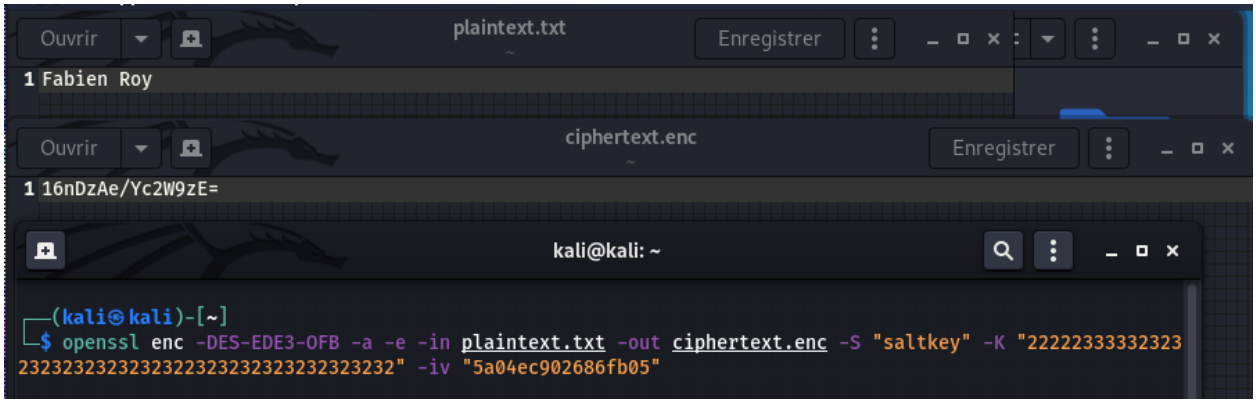
openssl enc -aes-256-cbc -a -nosalt -e -in plaintext.txt -out ciphertext.enc -K "2222233323233323223232323232232323323232323223232323232323222323" -iv "5a04ec902686fb05a6b7a338b6e07760"



Question 2

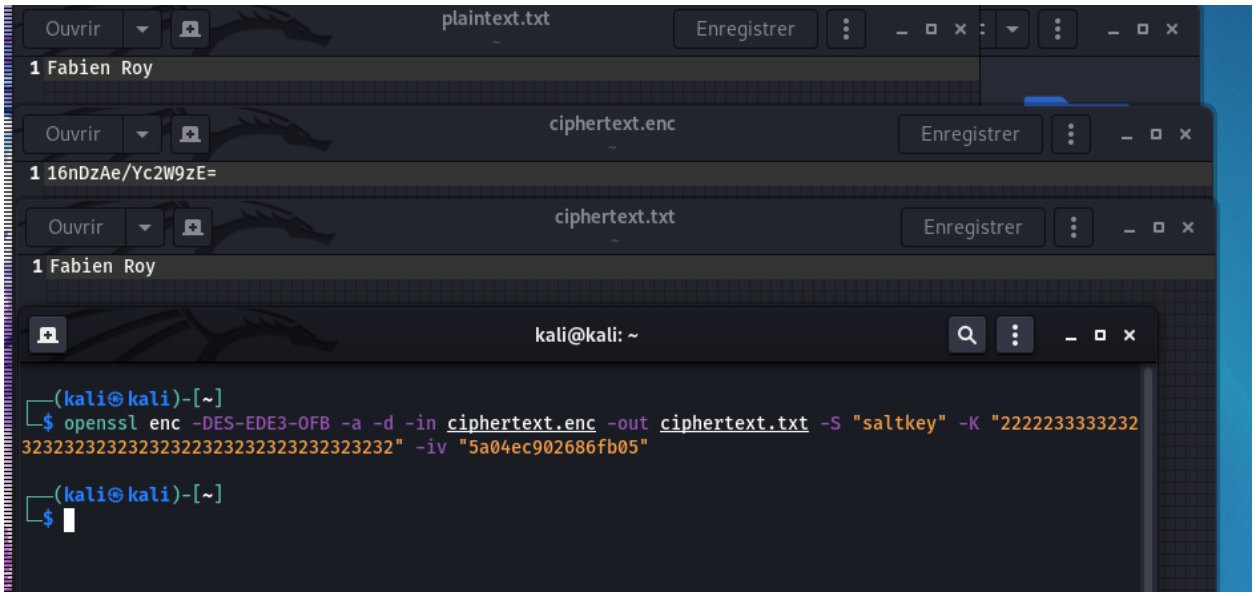
Encryption

openssl enc -DES-EDE3-OFB -a -e -in plaintext.txt -out ciphertext.enc -S "saltkey" -K "2222233333232323232323232323232232323232323232232" -iv "5a04ec902686fb05"



Decryption

openssl enc -DES-EDE3-OFB -a -d -in ciphertext.enc -out ciphertext.txt -S "saltkey" -K "2222233333232323232323232323232232323232323232323232" -iv "5a04ec902686fb05"



Question 3

Compression

tar -czvf plaintext.tar.gz plaintext.txt

Encryption

openssl enc -DESX-CBC -e -a -p -in plaintext.tar.gz -out ciphertext.txt

Mot de passe utilisé : “fabien”

```
Ouvrir                                ciphertext.txt
1 U2FsdGVkX18L+8v8YsIp+8ScXuHgfzh1QuCD31BwwqBQeoyuhSuriMk3ednxwKca
2 R9pfgEyJ2sG40s/JbYoUWcXOE+e/Xva00fY84sxJg+TbLR0TJJxt4PEVww8q0zZK
3 WWOqLyUUqfVpunH9Qz1VGRcxNaOTQMpSLHbR1LmML62AQIr/ENqfqx+yGOL0kQ3k
4 RbGLH+s8202aJK5EVRzMBw==

                                      kali@kali: ~

┌──(kali㉿kali)-[~]
└─$ openssl enc -DESX-CBC -e -a -p -in plaintext.tar.gz -out ciphertext.txt
enter desx-cbc encryption password:
Verifying - enter desx-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=A32D395CC7DA8450
key=26699E88103FD26BEEA46E5A3D21DCC023788F254CB31283
iv =A1248B7387839C01

┌──(kali㉿kali)-[~]
└─$
```

Question 4

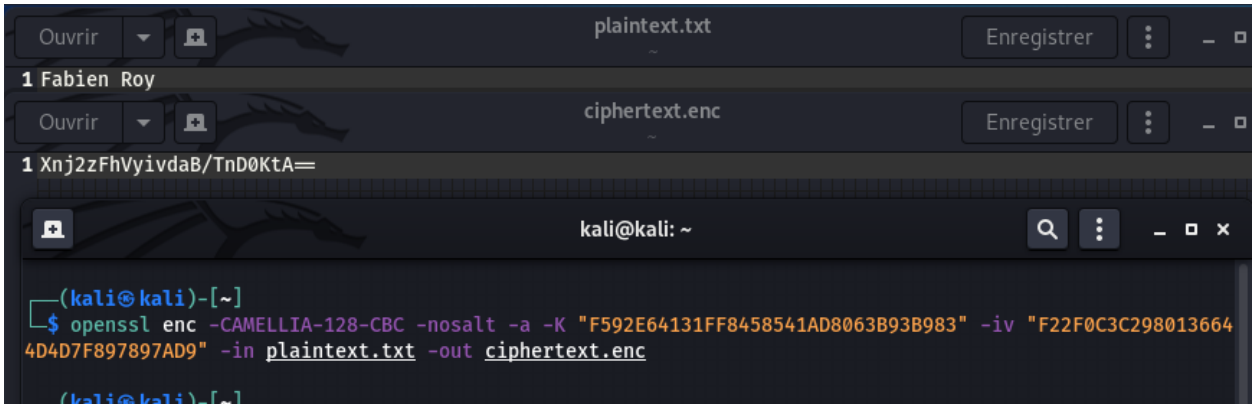
Génération de clé

openssl enc -aes-128-cbc -k fabien -P -md sha1

```
┌──(kali㉿kali)-[~]
└─$ openssl enc -aes-128-cbc -k fabien -P -md sha1
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=02B01E8DD5722FB3
key=F592E64131FF8458541AD8063B93B983
iv =F22F0C3C2980136644D4D7F897897AD9
```

Encryption

openssl enc -CAMELLIA-128-CBC -nosalt -a -K "F592E64131FF8458541AD8063B93B983" -iv "F22F0C3C2980136644D4D7F897897AD9" -in plaintext.txt -out ciphertext.enc

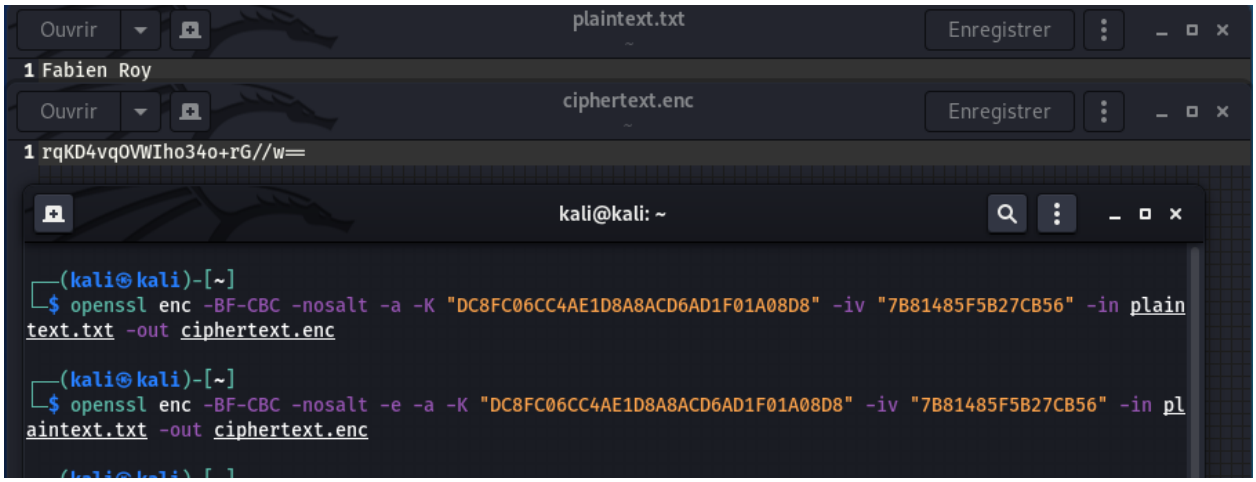


Question 5

Itération 1 : “Fabien Roy”

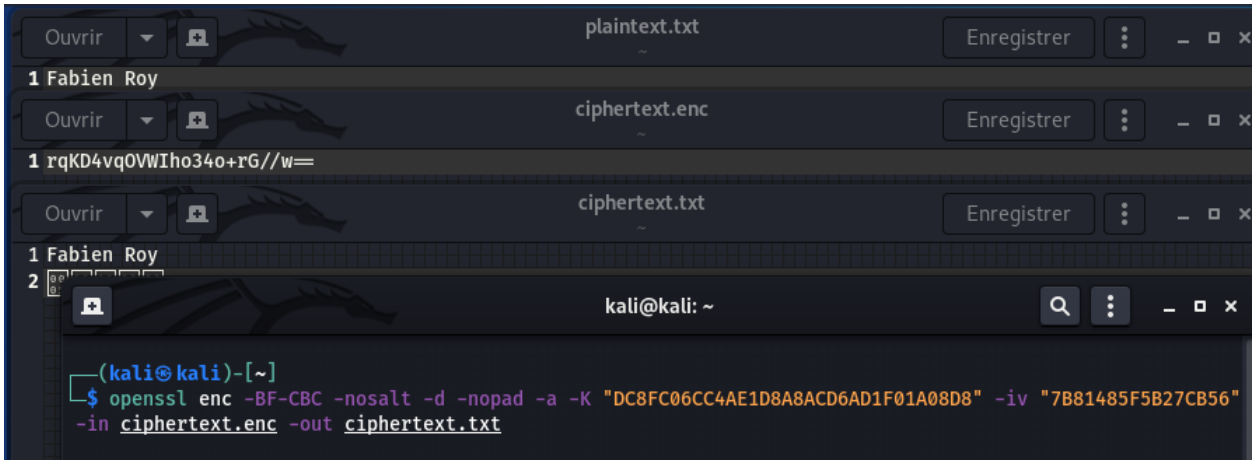
Encryption

openssl enc -BF-CBC -nosalt -e -a -K "DC8FC06CC4AE1D8A8ACD6AD1F01A08D8" -iv "7B81485F5B27CB56" -in plaintext.txt -out ciphertext.enc



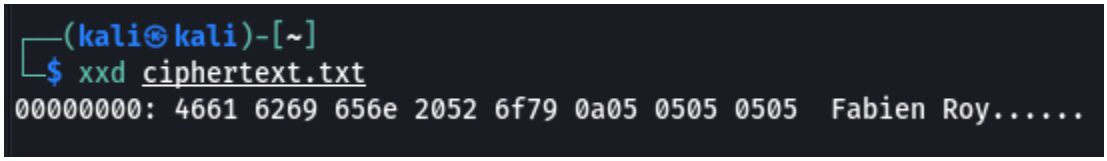
Decryption

openssl enc -BF-CBC -nosalt -d -nopad -a -K "DC8FC06CC4AE1D8A8ACD6AD1F01A08D8" -iv "7B81485F5B27CB56" -in ciphertext.enc -out ciphertext.txt



Visualisation

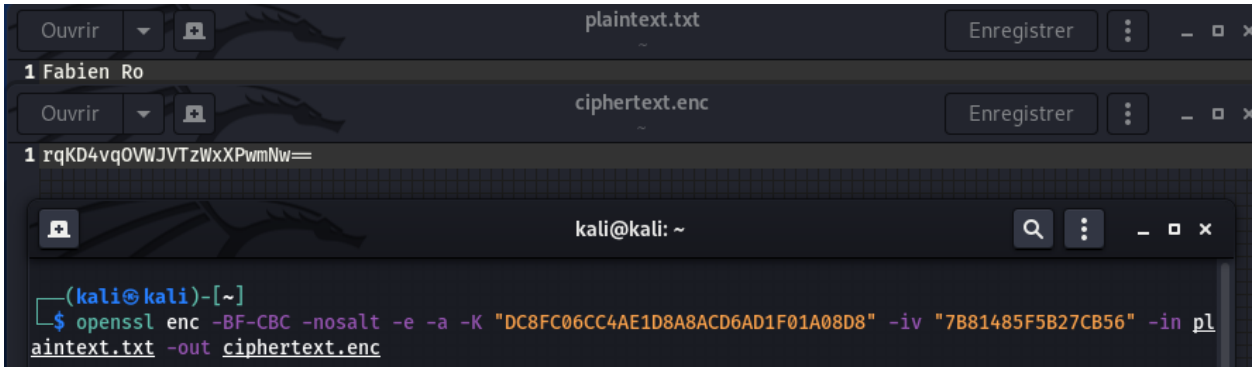
xxd ciphertext.txt



Itération 2 : “Fabien Ro”

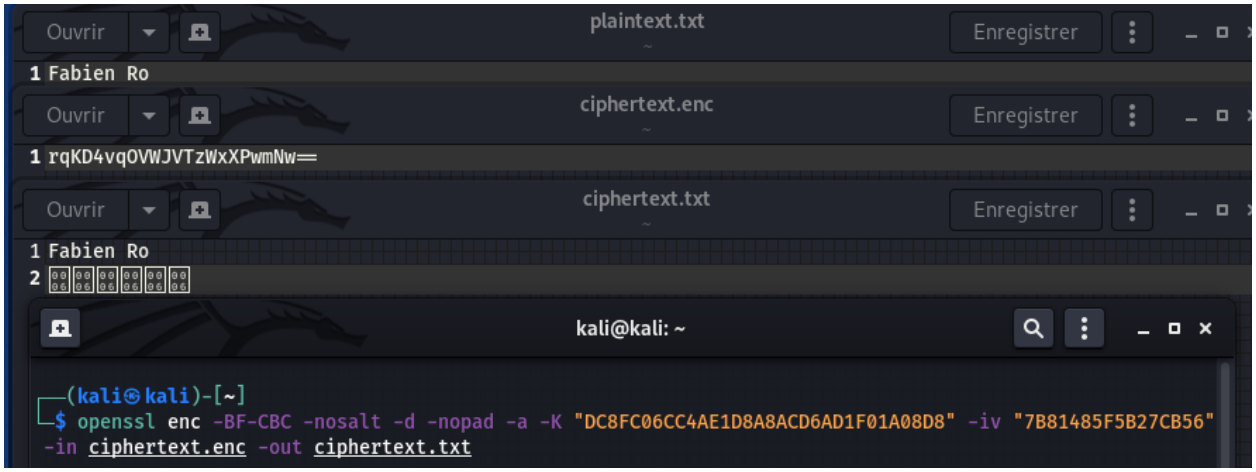
Encryption

openssl enc -BF-CBC -nosalt -e -a -K "DC8FC06CC4AE1D8A8ACD6AD1F01A08D8" -iv "7B81485F5B27CB56" -in plaintext.txt -out ciphertext.enc



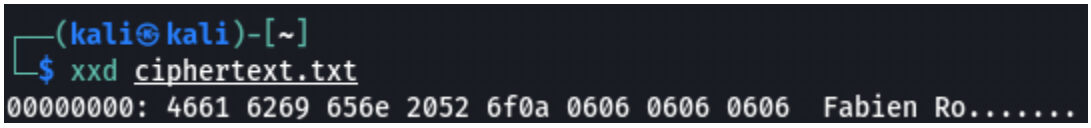
Decryption

openssl enc -BF-CBC -nosalt -d -nopad -a -K "DC8FC06CC4AE1D8A8ACD6AD1F01A08D8" -iv "7B81485F5B27CB56" -in ciphertext.enc -out ciphertext.txt



Visualisation

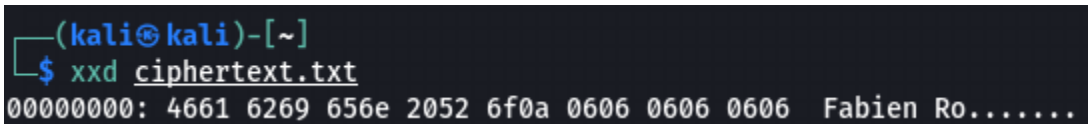
xxd ciphertext.txt



Itération 3 : “Fabien R”

Encryption

openssl enc -BF-CBC -nosalt -e -a -K "DC8FC06CC4AE1D8A8ACD6AD1F01A08D8" -iv "7B81485F5B27CB56" -in plaintext.txt -out ciphertext.enc



Decryption

openssl enc -BF-CBC -nosalt -d -nopad -a -K "DC8FC06CC4AE1D8A8ACD6AD1F01A08D8" -iv "7B81485F5B27CB56" -in ciphertext.enc -out ciphertext.txt

```
plaintext.txt
1 Fabien R
ciphertext.enc
1 rqKD4vqOVWI1UEB0DJqsdg==
ciphertext.txt
1 Fabien R
2 ⌧⌧⌧⌧⌧⌧⌧
kali@kali: ~
┌──(kali㉿kali)-[~]
└─$ openssl enc -BF-CBC -nosalt -d -nopad -a -K "DC8FC06CC4AE1D8A8ACD6AD1F01A08D8" -iv "7B81485F5B27CB56"
 -in ciphertext.enc -out ciphertext.txt
```

Visualisation

xxd ciphertext.txt

```
┌──(kali㉿kali)-[~]
└─$ xxd ciphertext.txt
00000000: 4661 6269 656e 2052 0a07 0707 0707 0707  Fabien R........
```

Question 6

AES par rapport à DES

AES

```
┌──(kali㉿kali)-[~]
└─$ openssl speed aes-128-cbc                                                    130 ✗
Doing aes-128 cbc for 3s on 16 size blocks: 54265264 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 64 size blocks: 13930863 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 256 size blocks: 3394480 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 1024 size blocks: 866371 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 8192 size blocks: 109296 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 16384 size blocks: 53930 aes-128 cbc's in 3.00s
OpenSSL 1.1.1k  25 Mar 2021
built on: Thu Mar 25 20:49:34 2021 UTC
options:bn(64,64) rc4(16x,int) des(int) aes(partial) blowfish(ptr)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -O2 -ffile-prefix-map=/bui
ld/openssl-4zpOOS/openssl-1.1.1k=. -fstack-protector-strong -Wformat -Werror=format-security -DOPENSSL_US
E_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENS
SL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECCAK1600_ASM -DRC4_ASM -DM
D5_ASM -DAESNI_ASM -DVPAES_ASM -DGHASH_ASM -DECP_NISTZ256_ASM -DX25519_ASM -DPOLY1305_ASM -DNDEBUG -Wdate
-time -D_FORTIFY_SOURCE=2
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes  16384 bytes
aes-128 cbc     289414.74k   297191.74k   289662.29k   295721.30k   298450.94k   294529.71k
```

On a donc :

- 16 bytes : 289414.74k

- 64 bytes : 297191.74k
- 256 bytes : 289662.29k
- 1024 bytes : 295721.30k
- 8192 bytes : 298450.94k
- 16384 bytes : 294529.71k

DES

```
┌──(kali㉿kali)-[~]
└─$ openssl speed aes-128-cbc                                                    130 ×
Doing aes-128 cbc for 3s on 16 size blocks: 54265264 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 64 size blocks: 13930863 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 256 size blocks: 3394480 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 1024 size blocks: 866371 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 8192 size blocks: 109296 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 16384 size blocks: 53930 aes-128 cbc's in 3.00s
OpenSSL 1.1.1k  25 Mar 2021
built on: Thu Mar 25 20:49:34 2021 UTC
options:bn(64,64) rc4(16x,int) des(int) aes(partial) blowfish(ptr)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -O2 -ffile-prefix-map=/bui
ld/openssl-4zpOOS/openssl-1.1.1k=. -fstack-protector-strong -Wformat -Werror=format-security -DOPENSSL_US
E_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENS
SL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECCAK1600_ASM -DRC4_ASM -DM
D5_ASM -DAESNI_ASM -DVPAES_ASM -DGHASH_ASM -DECP_NISTZ256_ASM -DX25519_ASM -DPOLY1305_ASM -DNDEBUG -Wdate
-time -D_FORTIFY_SOURCE=2
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes  16384 bytes
aes-128 cbc     289414.74k   297191.74k   289662.29k   295721.30k   298450.94k   294529.71k
```

On a donc :

- 16 bytes : 90673.62k
- 64 bytes : 93137.66k
- 256 bytes : 95435.35k
- 1024 bytes : 95859.03k
- 8192 bytes : 96490.84k
- 16384 bytes : 96321.54k

Comparaison

Après un bref calcul, on voit que DES est en moyenne 310.89% plus rapide que AES.

	AES	DES	DES/AES
16 bytes	289414,74	90673,62	319,18%
64 bytes	297191,74	93137,66	319,09%
256 bytes	289662,29	95435,35	303,52%
1024 bytes	295721,3	95859,03	308,50%
8192 bytes	298450,94	96490,84	309,30%
16384 bytes	294529,71	96321,54	305,78%
Moyenne			310,89%

DES par rapport à RSA

DES

```
┌──(kali㉿kali)-[~]
└─$ openssl speed aes-128-cbc                                                    130 ✗
Doing aes-128 cbc for 3s on 16 size blocks: 54265264 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 64 size blocks: 13930863 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 256 size blocks: 3394480 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 1024 size blocks: 866371 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 8192 size blocks: 109296 aes-128 cbc's in 3.00s
Doing aes-128 cbc for 3s on 16384 size blocks: 53930 aes-128 cbc's in 3.00s
OpenSSL 1.1.1k  25 Mar 2021
built on: Thu Mar 25 20:49:34 2021 UTC
options:bn(64,64) rc4(16x,int) des(int) aes(partial) blowfish(ptr)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -O2 -ffile-prefix-map=/bui
ld/openssl-4zpOOS/openssl-1.1.1k=. -fstack-protector-strong -Wformat -Werror=format-security -DOPENSSL_US
E_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENS
SL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECCAK1600_ASM -DRC4_ASM -DM
D5_ASM -DAESNI_ASM -DVPAES_ASM -DGHASH_ASM -DECP_NISTZ256_ASM -DX25519_ASM -DPOLY1305_ASM -DNDEBUG -Wdate
-time -D_FORTIFY_SOURCE=2
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes  16384 bytes
aes-128 cbc     289414.74k   297191.74k   289662.29k   295721.30k   298450.94k   294529.71k
```

On a donc :

- 16 bytes : 90673.62k
- 64 bytes : 93137.66k
- 256 bytes : 95435.35k
- 1024 bytes : 95859.03k
- 8192 bytes : 96490.84k
- 16384 bytes : 96321.54k

RSA

```
┌──(kali㉿kali)-[~]
└─$ openssl speed -seconds 3 rsa                                                    1 ✗
Doing 512 bits private rsa's for 3s: 90633 512 bits private RSA's in 3.00s
Doing 512 bits public rsa's for 3s: 1430900 512 bits public RSA's in 3.00s
Doing 1024 bits private rsa's for 3s: 42270 1024 bits private RSA's in 3.00s
Doing 1024 bits public rsa's for 3s: 630020 1024 bits public RSA's in 2.99s
Doing 2048 bits private rsa's for 3s: 4248 2048 bits private RSA's in 3.00s
Doing 2048 bits public rsa's for 3s: 71971 2048 bits public RSA's in 2.89s
Doing 3072 bits private rsa's for 3s: 1989 3072 bits private RSA's in 3.00s
Doing 3072 bits public rsa's for 3s: 99242 3072 bits public RSA's in 3.00s
Doing 4096 bits private rsa's for 3s: 910 4096 bits private RSA's in 3.00s
Doing 4096 bits public rsa's for 3s: 57246 4096 bits public RSA's in 3.00s
Doing 7680 bits private rsa's for 3s: 101 7680 bits private RSA's in 3.03s
Doing 7680 bits public rsa's for 3s: 17404 7680 bits public RSA's in 3.00s
Doing 15360 bits private rsa's for 3s: 20 15360 bits private RSA's in 3.15s
Doing 15360 bits public rsa's for 3s: 4657 15360 bits public RSA's in 3.00s
OpenSSL 1.1.1k  25 Mar 2021
built on: Thu Mar 25 20:49:34 2021 UTC
options:bn(64,64) rc4(16x,int) des(int) aes(partial) blowfish(ptr)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -Wa,--noexecstack -g -O2 -ffile-prefix-map=/build/openssl
-4zpOOS/openssl-1.1.1k=. -fstack-protector-strong -Wformat -Werror=format-security -DOPENSSL_USE_NODELETE -DL_ENDIA
N -DOPENSSL_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_A
SM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECCAK1600_ASM -DRC4_ASM -DMD5_ASM -DAESNI_ASM -DVPAES_ASM -DGHASH_A
SM -DECP_NISTZ256_ASM -DX25519_ASM -DPOLY1305_ASM -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=2
                  sign    verify    sign/s verify/s
rsa  512 bits 0.000033s 0.000002s  30211.0 476966.7
rsa 1024 bits 0.000071s 0.000005s  14090.0 210709.0
rsa 2048 bits 0.000706s 0.000040s   1416.0  24903.5
rsa 3072 bits 0.001508s 0.000030s    663.0  33080.7
rsa 4096 bits 0.003297s 0.000052s    303.3  19082.0
rsa 7680 bits 0.030000s 0.000172s     33.3   5801.3
rsa 15360 bits 0.157500s 0.000644s      6.3   1552.3
```

On a donc, en moyenne, pour la signature :

- 512 bits : 0.000033s
- 1024 bits : 0.000071s
- 2048 bits : 0.000706s
- 3072 bits : 0.001508s
- 4096 bits : 0.003297s
- 7680 bits : 0.030000s
- 15360 bits : 0.157500s

Comparaison

Après un bref calcul, on voit que RSA est en moyenne 18268.77% plus rapide que DES.

	Bytes	DES (enc)	Bytes * DES		Bits	RSA (s)	Bits / RSA (bit/s)
	16	90673,62	1450777,92		512	0,000033	15515151,52
	64	93137,66	5960810,24		1024	0,000071	14422535,21
	256	95435,35	24431449,6		2048	0,000706	2900849,858
	1024	95859,03	98159646,72		3072	0,001508	2037135,279
	8192	96490,84	790452961,3		4096	0,003297	1242341,523
	16384	96321,54	1578132111		7680	0,03	256000
	Moyenne par byte		416431292,9		15360	0,1575	97523,80952
	Moyenne par bit		3331450343		Moyenne par bit/s		6078589,533
(3 secondes)	Moyenne par bit/s		1110483448				
			DES/RSA	18268,77%			