

Devoir : Mise en œuvre d'une PKI et sécurisation de flux critiques avec OpenSSL

Génération d'une clé privée, d'une clé publique et demande de signature de certificat pour Alice :

```
user@DEBIAN-01:~/alice$ openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -pkeyopt rsa_keygen_
_pubexp:3 -out privkey-Alice.pem
.....+-----+
+-----+*.....+
.....+-----+*.....+
.....+-----+*.....+
.....+-----+*.....+
user@DEBIAN-01:~/alice$ openssl pkey -in privkey-Alice.pem -pubout -out pubkey-Alice.pem
user@DEBIAN-01:~/alice$ openssl req -new -key privkey-Alice.pem -out Alice-req.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:IDF
Locality Name (eg, city) []:Melun
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CCI
Organizational Unit Name (eg, section) []:UTEC
Common Name (e.g. server FQDN or YOUR name) []:Alice
Email Address []:alice@utec.fr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:alice
An optional company name []:utec
```

Génération d'une clé privée, d'une clé publique et demande de signature de certificat pour Bob :

```
user@DEBIAN-01:~/bob$ openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -pkeyopt rsa_keygen_
ubexp:3 -out privkey-Bob.pem
.....+-----+
+-----+*.....+
.....+-----+*.....+
.....+-----+*.....+
.....+-----+*.....+
user@DEBIAN-01:~/bob$ openssl pkey -in privkey-Bob.pem -pubout -out pubkey-Bob.pem
user@DEBIAN-01:~/bob$ openssl req -new -key privkey-Bob.pem -out Bob-req.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:IDF
Locality Name (eg, city) []:Melun
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CCI
Organizational Unit Name (eg, section) []:UTEC
Common Name (e.g. server FQDN or YOUR name) []:Bob
Email Address []:bob@utec.fr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:boby
An optional company name []:utec
```

Générer un certificat auto-signé pour l'AC :

```
user@DEBIAN-01:~/ac$ openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -pkeyopt rsa_keygen_pubexp:3 -out privkey-AC.pem
.....
user@DEBIAN-01:~/ac$ openssl pkey -in privkey-AC.pem -pubout -out pubkey-AC.pem
user@DEBIAN-01:~/ac$ openssl req -x509 -new -nodes -key privkey-AC.pem -sha256 -days 3650 -out AC.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:IDF
Locality Name (eg, city) []:Melun
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CCI
Organizational Unit Name (eg, section) []:UTEC
Common Name (e.g. server FQDN or YOUR name) []:utec.fr
Email Address []:admin@utec.fr
```

A ce stade, on peut générer et signer des certificats pour Alice et Bob avec l'AC :

Générer et signer un certificat pour Alice avec l'AC :

```
user@DEBIAN-01:~/alice$ openssl x509 -req -in Alice-req.csr -CA ../ac/AC.crt -CAkey ../ac/privkey-AC.pem  
-CAcreateserial -out Alice.crt -days 500 -sha256  
Certificate request self-signature ok  
subject=C=FR, ST=IDF, L=Melun, O=CCI, OU=UTEC, CN=Alice, emailAddress=alice@utec.fr
```

Générer et signer un certificat pour Bob avec l'AC :

```
user@DEBIAN-01:~/bob$ openssl x509 -req -in Bob-req.csr -CA ../ac/AC.crt -CAkey ../ac/privkey-AC.pem -CA
createserial -out Bob.crt -days 500 -sha256
Certificate request self-signature ok
subject=C=FR, ST=IDF, L=Melun, O=CCI, OU=UTEC, CN=Bob, emailAddress=bob@utec.fr
```

Alice vérifie le certificat public de Bob :

```
user@DEBIAN-01:~/alice$ openssl verify -CAfile ../ac/AC.crt ../bob/Bob.crt
../bob/Bob.crt: OK
```

Bob vérifie le certificat public de Alice :

```
user@DEBIAN-01:~/bob$ openssl verify -CAfile ../ac/AC.crt ../alice/Alice.crt
../alice/Alice.crt: OK
```


Extraction de la clé publique de Bob par Alice :

```
user@DEBIAN-01:~/alice$ openssl x509 -pubkey -in ../bob/Bob.crt -noout > pubkey-Bob.pem -pubkey
user@DEBIAN-01:~/alice$ ls -l
total 20
-rw-rw-r-- 1 user user 1078 Mar 27 13:09 Alice-req.csr
-rw-rw-r-- 1 user user 1354 Mar 27 13:40 Alice.crt
-rw----- 1 user user 1704 Mar 27 13:08 privkey-Alice.pem
-rw-rw-r-- 1 user user 451 Mar 27 13:08 pubkey-Alice.pem
-rw-rw-r-- 1 user user 451 Mar 27 13:44 pubkey-Bob.pem
```

Création du fichier « rapport_audit.pdf » de 15Mo :

```
user@DEBIAN-01:~/alice$ head -c 15M </dev/urandom > rapport_audit.pdf
user@DEBIAN-01:~/alice$ ls -lh rapport_audit.pdf
-rw-rw-r-- 1 user user 15M Mar 27 13:45 rapport_audit.pdf
```

Tentative de de chiffrement du fichier « rapport_audit.pdf » avec la clé publique de Bob :

```
user@DEBIAN-01:~/alice$ openssl pkeyutl -encrypt -in rapport_audit.pdf -pubin -inkey pubkey-Bob.pem -out
ciphertext.bin
Public Key operation error
4017FBCF6C7F0000:error:0200006E:rsa routines:ossl_rsa_padding_add_PKCS1_type_2_ex:data too large for key
size:../crypto/rsa/rsa_pk1.c:132:
```

Explication technique : Pourquoi ne peut-on pas chiffrer directement le rapport de 15 Mo avec la clé publique de Bob ?

L'algorithme RSA est mathématiquement incapable de chiffrer des données plus grandes que sa propre taille de clé. La clé utilisé fait 2048 bits, soit 256 octets, le fichier « rapport_audit.pdf » que l'on tente de chiffrer fait 15728640 octets.

Génération de clé symétrique, Alice chiffre symkey.pem en utilisant la clé publique de Bob et Alice hache symkey.pem et le chiffre à l'aide de sa clé privée :

```
user@DEBIAN-01:~/alice$ openssl rand -base64 -out symkey.pem 32
user@DEBIAN-01:~/alice$ openssl pkeyutl -encrypt -in symkey.pem -pubin -inkey pubkey-Bob.pem -out symkey
.enc
user@DEBIAN-01:~/alice$ openssl dgst -sha1 -sign privkey-Alice.pem -out signature.bin symkey.pem
```

A ce stade, voici l'infrastructure d'Alice :

```
user@DEBIAN-01:~/alice$ ls -l
total 15392
-rw-rw-r-- 1 user user 1078 Mar 27 13:09 Alice-req.csr
-rw-rw-r-- 1 user user 1354 Mar 27 13:40 Alice.crt
-rw-rw-r-- 1 user user 0 Mar 27 13:48 ciphertext.bin
-rw----- 1 user user 1704 Mar 27 13:08 privkey-Alice.pem
-rw-rw-r-- 1 user user 451 Mar 27 13:08 pubkey-Alice.pem
-rw-rw-r-- 1 user user 451 Mar 27 13:44 pubkey-Bob.pem
-rw-rw-r-- 1 user user 15728640 Mar 27 13:45 rapport_audit.pdf
-rw-rw-r-- 1 user user 256 Mar 27 13:52 signature.bin
-rw-rw-r-- 1 user user 256 Mar 27 13:51 symkey.enc
-rw-rw-r-- 1 user user 45 Mar 27 13:50 symkey.pem
```

Bob vérifie que le message provient d'Alice :

```
user@DEBIAN-01:~/bob$ openssl x509 -pubkey -in ../alice/Alice.crt -noout > pubkey-Alice.pem
user@DEBIAN-01:~/bob$ openssl dgst -sha1 -verify pubkey-Alice.pem -signature signature.bin symkey.pem
Verified OK
```

Alice chiffre son rapport_audit.pdf avec la clé symétrique et envoie rapport_audit.pdf.enc à Bob :

```
user@DEBIAN-01:~/alice$ cp symkey.enc signature.bin ../bob/
user@DEBIAN-01:~/alice$ openssl enc -aes-256-cbc -pass file:symkey.pem -p -md sha256 -pbkdf2 -iter 10000
0 -in rapport_audit.pdf -out rapport_audit.pdf.enc
salt=BC497C5651864343
key=955D8601BE56E98B93E4549B9879422EEDD4B0AB460C2CA79AEE7BF500E11958
iv =E698F212232D47EBE40D9F3146252D85
user@DEBIAN-01:~/alice$ cp rapport_audit.pdf.enc ../bob/
```

Bob déchiffre rapport_audit.pdf.enc avec la même clé symétrique :

```
user@DEBIAN-01:~/bob$ openssl dgst -sha1 -verify pubkey-Alice.pem -signature signature.bin symkey.pem
Verified OK
user@DEBIAN-01:~/bob$ openssl enc -aes-256-cbc -d -pass file:symkey.pem -p -md sha256 -pbkdf2 -iter 10000
00 -in rapport_audit.pdf.enc -out rapport_audit.pdf -d
salt=BC497C5651864343
key=955D8601BE56E98B93E4549B9879422EEDD4B0AB460C2CA79AEE7BF500E11958
iv =E698F212232D47EBE40D9F3146252D85
```

Preuve d'intégrité : Fournissez les **empreintes SHA-256** du **fichier original** et du **fichier final déchiffré**.

Voici les « sha256sum » du fichier chiffré et envoyer par Alice et du fichier déchiffré que Bob à reçu :

```
user@DEBIAN-01:~/bob$ sha256sum rapport_audit.pdf
9f8cad0db8d7d883405cb6b6dcfc5a0c68597e1bbfca2417bd3e11c5ee37e785 rapport_audit.pdf
user@DEBIAN-01:~/bob$ sha256sum ../alice/rapport_audit.pdf
9f8cad0db8d7d883405cb6b6dcfc5a0c68597e1bbfca2417bd3e11c5ee37e785 ../alice/rapport_audit.pdf
```

Analyse de sécurité : Quel est l'impact de l'utilisation de l'**exposant e=3** par rapport au standard **e=65537** ?

L'utilisation de $e=3$ au lieu de $e=65537$ divise par huit le travail de calcul pour chiffrer un message, ce qui profite aux appareils très lents comme les cartes à puce. En revanche, ce gain de vitesse fragilise énormément la sécurité car une simple racine cubique mathématique peut suffire à briser le secret si le message est trop court ou mal protégé. Le standard 65537 est le "nombre d'or" car il reste extrêmement rapide pour un processeur moderne tout en bloquant nativement ces attaques par diffusion qui touchent les petits exposants. En résumé, on sacrifie une marge de sécurité massive pour un gain de performance qui n'est plus du tout nécessaire sur les ordinateurs d'aujourd'hui.

Voici l'infrastructure final de la configuration correcte de l'AC et des certificats :

```
*  
|-- ac  
|  |-- AC.crt  
|  |-- AC.srl  
|  |-- privkey-AC.pem  
|  `-- pubkey-AC.pem  
|-- alice  
|  |-- Alice-req.csr  
|  |-- Alice.crt  
|  |-- ciphertext.bin  
|  |-- privkey-Alice.pem  
|  |-- pubkey-Alice.pem  
|  |-- pubkey-Bob.pem  
|  |-- rapport_audit.pdf  
|  |-- rapport_audit.pdf.enc  
|  |-- signature.bin  
|  |-- symkey.enc  
|  `-- symkey.pem  
|-- bob  
|  |-- Bob-req.csr  
|  |-- Bob.crt  
|  |-- privkey-Bob.pem  
|  |-- pubkey-Alice.pem  
|  |-- pubkey-Bob.pem  
|  |-- rapport_audit.pdf  
|  |-- rapport_audit.pdf.enc  
|  |-- signature.bin  
|  |-- symkey.enc  
|  `-- symkey.pem
```