Concepts de sécurité informatique

Arnaud Grandville
14 décembre 2009

Introduction

Société de l'information
->communications
->l'information est un capital (fichier client, numéros de cartes bancaires, …)

Vulnérabilités :
- Applications
- Système
- Réseau

Limiter
- les informations exposées
- Limiter les privilèges
- Limiter les services
- Limiter les protocoles

Défense en profondeur, ne pas se reposer sur la sécurité de l'élément précédent pour assurer sa propre défense -> Architecture trois tiers, équipements réseau (Firewall, routeurs avec ACL)

Serveurs LAMP (linux, Apache, mySQL et PHP)
Linux
Apache
Serveurs mySQL
PHP

Cryptographie

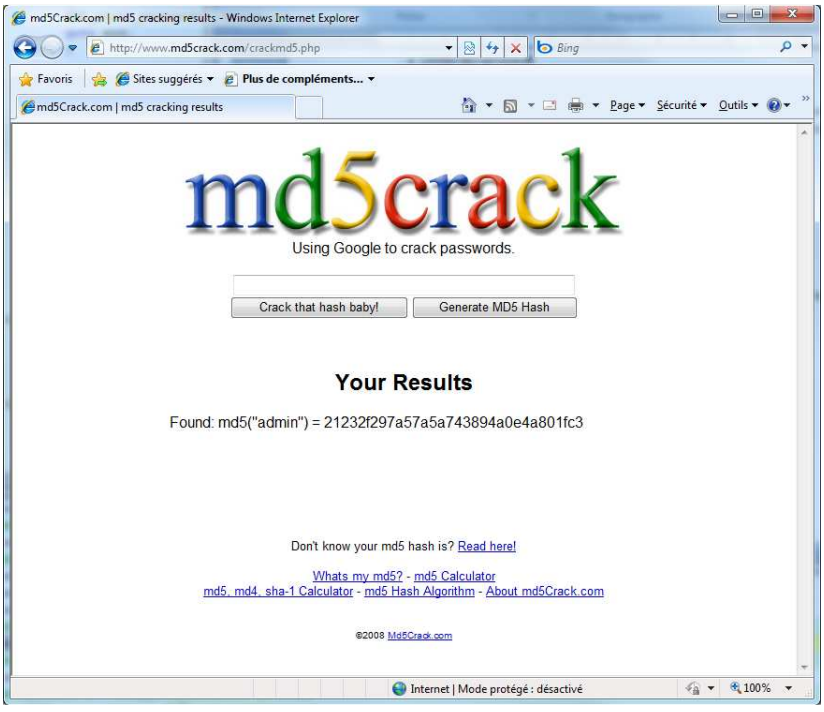
hashage , encodage, chiffrement Symétriques, chiffrement Asymétriques

Hashage

compression pour obtenir une signature, utilisation d'une valeur d'initialisation (salt) pour éviter les attaques par dictionnaire.

MD5 (1991) et cracké en 1995, 128 bits (32 caractères hexadécimaux)
Nota : 0->F hexa/0000->1111 Binaire)

```
# echo -n admin | openssl md5
21232f297a57a5a743894a0e4a801fc3
# echo -n admin | openssl dgst -md5
21232f297a57a5a743894a0e4a801fc3
# echo -n admiN | openssl dgst -md5
806d8e1e323557a073a79d006c3fae11
# echo -n admiN | openssl dgst -md5 -binary -out md5.bin
# openssl dgst -hex md5.bin
MD5(md5.bin)= 2c2fcba8df4f56caf7ed084eede41bf9
```



SHA1 (1995) Collisions en 2^80 itérations soit quelques semaines de calculs, 160 bits (40 caractères)

echo -n admin | openssl sha1
d033e22ae348aeb5660fc2140aec35850c4da997

variantes -> SHA256,384 et 512 bits

Encodages

Hexadécimal

H	D	B
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

8D0=8x16^2+13*16+0=2256

16^4=65536
16^3=4096
16^2=256

5023-1x16^3=927
927-3x16^2=159
159-9x16=15
15
->139F

Base64

6 bits par caractères (2^6 = table de 64 caractères + un complément)

0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H
8	I
9	J
10	K
11	L
12	M
13	N
14	O
15	P
16	Q
17	R
18	S
19	T
20	U
21	V

22	W
23	X
24	Y
25	Z
26	a
27	b
28	c
29	d
30	e
31	f
32	g
33	h
34	i
35	J
36	k
37	l
38	m
39	n
40	o
41	p
42	q
43	r

44	s
45	t
46	u
47	v
48	w
49	x
50	y
51	z
52	0
53	1
54	2
55	3
56	4
57	5
58	6
59	7
60	8
61	9
62	+
63	/
	=

La sortie est toujours un multiple de 4 caractères, la longueur de la sortie est augmentée de 8/6

a	d	m	i				
97	100	109	105				
01100001	01100100	01101101	01101001				
011000	010110	010001	101101	011010	010000		
24	22	17	45	26	16		
Y	W	R	T	a	Q	=	=

a	d	m	i	n			
97	100	109	105	110			
01100001	01100100	01101101	01101001	01101110			
011000	010110	010001	101101	011010	010110	111000	
24	22	17	45	26	22	56	
Y	W	R	T	a	W	4	=

Les chiffrements

Chiffre de César

Chiffre de Vigenere (1523-1596)

Chiffre de Beale (1820, Virginie) Thomas J Beale séjour deux années dans un hôtel de Lynchburg et confie au propriétaire avec qui il s'était lié d'amitié une boite métallique. M. Morriss le propriétaire reçoit quelques semaines plus tard un courrier lui indiquant que le contenu de cette boite renferme des informations sur la fortune de M. Beale et de ses camarades, avec comme consigne que si personne ne vient lui réclamer d'ici à dix ans, il devra l'ouvrir. La boite contient trois lettres chiffrées incompréhensibles sans la clef qu'un ami doit poster à partir de juin 1832 ni le courrier, ni personne ne viendra et la boite restera fermée jusqu'en 1845 date à laquelle, il découvre une lettre en clair précisant l'origine de cette fortune et les trois courriers chiffrés indiquant l'emplacement du trésor, l'inventaire et la liste des bénéficiaires.
En 1862, M. Morriss confie cette histoire à un ami qui la rendra publique en 1885 après avoir déchiffré le deuxième courrier et avoir testé tous les textes qui lui passent sous la main jusqu'au jour ou il l'a déchiffré avec la déclaration d'indépendance des États-Unis d'Amérique.

Tartuffe
Comment ? Couvrez ce sein que je ne saurais voir.
Par de pareils objets les âmes sont blessées,
Et cela fait venir de coupables pensées.

1	C
2	C
3	C
4	S
5	Q
6	J
7	N
8	S
9	V
10	P
11	D
12	P

13	O
14	L
15	A
16	S
17	B
18	E
19	C
20	F
21	V
22	D
23	C
24	P

CODES
1,13,11,18,4

Machine Enigma

Alan Turing (1912-1954), 1970 révélation de sa participation, 2009 présentation des regrets du gouvernement britannique.

Chiffrements symétriques

```
# openssl enc ?
unknown option '?'
options are
-in <file>     input file
-out <file>    output file
-pass <arg>    pass phrase source
-e             encrypt
-d             decrypt
-a/-base64     base64 encode/decode, depending on encryption flag
-k             passphrase is the next argument
-kfile         passphrase is the first line of the file argument
-md            the next argument is the md to use to create a key
                 from a passphrase.  One of md2, md5, sha or sha1
-K/-iv         key/iv in hex is the next argument
-[pP]          print the iv/key (then exit if -P)
-bufsize <n>   buffer size
-engine e      use engine e, possibly a hardware device.
Cipher Types
-aes-128-cbc               -aes-128-cfb               -aes-128-cfb1
-aes-128-cfb8              -aes-128-ecb               -aes-128-ofb
-aes-192-cbc               -aes-192-cfb               -aes-192-cfb1
-aes-192-cfb8              -aes-192-ecb               -aes-192-ofb
-aes-256-cbc               -aes-256-cfb               -aes-256-cfb1
-aes-256-cfb8              -aes-256-ecb               -aes-256-ofb
-aes128                    -aes192                    -aes256
-bf                        -bf-cbc                    -bf-cfb
-bf-ecb                    -bf-ofb                    -blowfish
-cast                      -cast-cbc                  -cast5-cbc
-cast5-cfb                 -cast5-ecb                 -cast5-ofb
-des                       -des-cbc                   -des-cfb
-des-cfb1                  -des-cfb8                  -des-ecb
-des-ede                   -des-ede-cbc               -des-ede-cfb
-des-ede-ofb               -des-ede3                  -des-ede3-cbc
-des-ede3-cfb              -des-ede3-ofb              -des-ofb
-des3                      -desx                      -desx-cbc
-rc2                       -rc2-40-cbc                -rc2-64-cbc
-rc2-cbc                   -rc2-cfb                   -rc2-ecb
-rc2-ofb                   -rc4                       -rc4-40
```

Nécessité d'un salt !

ECB, chaque bloc est chiffré indépendamment de son voisin
Deux blocs identiques donnent le même résultat
CBC, Cipher Block Chaining : Le bloc suivant est conjugué avec le courant,
E(Texte⊕ Texte Chiffré précédent) ⇒ texte chiffré
CFB, Cipher FeedBack
E(Clef+bloc chiffrée précédent) ⊕Texte⇒ texte chiffré
OFB, Output feedback
E(Clef+Clef du bloc précédent) ⊕Texte⇒ texte chiffré

(56 bits,112,168)
des (1977)
des-ede -> double des
des-ede3 -> tripe des

RC (Ronald Rivest) Adi Shamir et Den Adleman, Ron's Code ou Rivest Cipher
RC2,RC4 (wep) 1987 et RC5 (1994)

Blowfish 1993 (32 à 448 bits)
AES 1997, Rijndael (Joan Daemen et Vincent Rijmen)

Chiffrement asymétrique

principes :

Ne pas déduire la clef privée de la clef publique
Ne pas décoder le texte crypté

Rivest Shamir Adleman (1977)

p et q deux nombres premiers
n=p*q
calculer l'indicatrice d'euler φ (n) (nombre de nombres de premiers inférieurs à n)= (p-1)*(q-1)
choisir e tel que e et φ (n) soient premiers entre eux
Théorème de Bachet-Bézout -> e*d ≡1 mod (φ (n))
$d=e^{-1}$ mod (φ (n))

chiffrement
$c=m^e$*mod n

Déchiffrement
$m=c^d$*mod n

(n,e) = clef publique
(n,d) = clef privée

exemple
p=47
q=71
n=p*q=3337
choisir e sans qu'il ai de facteur commun avec 46*70 = 3320
e=79 parce que premier et ne divise pas 3320.
79*x=1+ 3220*y

```
x=1
do
cal = ((x*3220)+1)/79
wscript.echo x&" "&cal
x=x+1
loop while(cal<>round(cal))
```

1 40,7721518987342
2 81,5316455696203
3 122,291139240506
4 163,050632911392
5 203,810126582278
6 244,569620253165
7 285,329113924051
8 326,088607594937
9 366,848101265823
10 407,607594936709
11 448,367088607595

12 489,126582278481
13 529,886075949367
14 570,645569620253
15 611,405063291139
16 652,164556962025
17 692,924050632911
18 733,683544303797
19 774,443037974684
20 815,20253164557
21 855,962025316456
22 896,721518987342
23 937,481012658228
24 978,240506329114
25 1019

d=1019
((79*1019)-1)/3220=25

```
x=1
do
cal = ((x*3220)+1) mod 79
wscript.echo x&" "&cal
x=x+1
loop while(cal<>0)
```

algorithme d'euclide pour la détermination de x et y simultanément !
Phase 1
3220 = 79 * 40 + 60
79 = 1 * 60 + 19
60 = 3 * 19 + 3
19 = 6 * 3 + 1
Phase 2
1 = 19 – (6 * 3) et 3 = 60 – (3 * 19) => 1=19-6*(60-3*19) =(1+18)*19-6*60
19=79-1*60 => 1=19*(79-1*60)-6*60=19*79-25*60
60=3220-79*40=> 1=19*79-25*(3220-79*40)=(19+1000)*79-25x3220

Message = 6882326879666683
M1=688
M2=232
M3=687
M4=966
M5=668
M6=003

$C1=688^{79} \bmod 3337=1570$
C2=2756
C3=2091
C4=2276
C5=2423
C6=158

$C1^{1019} \bmod 3337=688$

```
x=688
y=79
m=3337
cal=1

for a=1 to y
cal=(cal*x) mod m
next
wscript.echo "="&cal
wscript.quit(0)
```

$a^8 \bmod n$
$=(a*a*a*a*a*a*a*a) \bmod n$
$=((a^2 \bmod n)^2 \bmod n)^2 \bmod n$

Commutative, l'ordre des termes n'a pas d'influence sur le résultat
Associative,pas de priorité de calcul (parenthèses)
Distributive, on passe d'un produit de sommes à une somme de produit

$(a+b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
$(a-b) \bmod n = ((a \bmod n) – (b \bmod n)) \bmod n$
$(a*b) \bmod n = ((a \bmod n) * (b \bmod n)) \bmod n$
$(a*(b+c)) \bmod n = (((a*b) \bmod n)+((a*c) \bmod n)) \bmod n$
$a^b*a^c=a^{(b+c)}$
$(a^b)^c=a^{b*c}$

$a^{25} \bmod n$
$= (a*a^{24}) \bmod n$
$= (a*a^8 *a^{16}) \bmod n$
$=(a*((a^2)^2)^2*(((a^2)^2)^2)^2 \bmod n$
$=((((a^2*a)^2)^2)^2*a) \bmod n$
$=(((((((a^2 \bmod n) *a) \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n)*a) \bmod n$

Les certificats

Formats PEM,DER(ASN.1)

Les étapes de création d'un certificat

Génération des clefs

Génération de la certificat request

Signature de la requête par l'autorité

Openssl
Variable d'environnement
OPENSSL_CONF

GNU Multiple Precision Arithmetic Library
http://gmplib.org/

ASN.1

Abstract Syntax Notation One

```
Example { 1 2 3 4 }
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
	Group ::= SEQUENCE {
		id	OBJECT IDENTIFIER,
		value	Value
	}
	Value ::= SEQUENCE {
		value1	INTEGER,
		value2	BOOLEAN
	}
END
```

accès à la valeur Example.Group.id

Définition ASN1 d'un certificat (http://www.ietf.org/rfc/rfc2459.txt §4.1)

```
Certificate  ::=  SEQUENCE  {
	tbsCertificate		TBSCertificate,
	signatureAlgorithm	AlgorithmIdentifier,
	signatureValue		BIT STRING
	}

TBSCertificate  ::=  SEQUENCE  {
	version		[0]  EXPLICIT Version DEFAULT v1,
	serialNumber		CertificateSerialNumber,
	signature		AlgorithmIdentifier,
	issuer			Name,
	validity		Validity,
	subject			Name,
	subjectPublicKeyInfo SubjectPublicKeyInfo,
	issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
	-- If present, version shall be v2 or v3
	subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
	-- If present, version shall be v2 or v3
	extensions	[3]  EXPLICIT Extensions OPTIONAL
	-- If present, version shall be v3
	}

Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }

CertificateSerialNumber  ::=  INTEGER

Validity ::= SEQUENCE {
	notBefore	Time,
	notAfter	Time
	}

Time ::= CHOICE {
	utcTime		UTCTime,
	generalTime	GeneralizedTime
	}

UniqueIdentifier  ::=  BIT STRING

SubjectPublicKeyInfo  ::=  SEQUENCE  {
	algorithm		AlgorithmIdentifier,
	subjectPublicKey	BIT STRING
	}

Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension

Extension  ::=  SEQUENCE  {
	extnID		OBJECT IDENTIFIER,
	critical	BOOLEAN DEFAULT FALSE,
	extnValue	OCTET STRING
```

```
        }

AlgorithmIdentifier  ::=  SEQUENCE  {
        algorithm               OBJECT IDENTIFIER,
        parameters              ANY DEFINED BY algorithm OPTIONAL
}

Name ::= CHOICE {
        RDNSequence
        }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
        type     AttributeType,
        value    AttributeValue
        }

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType
```

```
C:\OpenSSL>openssl asn1parse -in ebay.crt -dump
    0:d=0  hl=4 l=1865 cons: SEQUENCE
    4:d=1  hl=4 l=1585 cons: SEQUENCE
    8:d=2  hl=2 l=   3 cons: cont [ 0 ]
   10:d=3  hl=2 l=   1 prim: INTEGER           :02
   13:d=2  hl=2 l=  16 prim: INTEGER           :3B32696932581E33D0EEE52A19702775

   31:d=2  hl=2 l=  13 cons: SEQUENCE
   33:d=3  hl=2 l=   9 prim: OBJECT            :sha1WithRSAEncryption
   44:d=3  hl=2 l=   0 prim: NULL
   46:d=2  hl=3 l= 186 cons: SEQUENCE
   49:d=3  hl=2 l=  11 cons: SET
   51:d=4  hl=2 l=   9 cons: SEQUENCE
   53:d=5  hl=2 l=   3 prim: OBJECT            :countryName
   58:d=5  hl=2 l=   2 prim: PRINTABLESTRING   :US
   62:d=3  hl=2 l=  23 cons: SET
   64:d=4  hl=2 l=  21 cons: SEQUENCE
   66:d=5  hl=2 l=   3 prim: OBJECT            :organizationName
   71:d=5  hl=2 l=  14 prim: PRINTABLESTRING   :VeriSign, Inc.
```

Génération des clefs
OPENSSL GENRSA

Paramètres courants de la ligne de commande	
-des,-des2,-idea,aes128,-aes192,-aes256	Protection de la clef
-out	Fichier de sortie
-passout	Mot de passe -<nom>=pass :<valeur> -<nom>=env:<valeur> -<nom>=file:<valeur> -<nom>=fd:<valeur> -<nom>=stdin
-rand file ;file	Source de hasard (à défaut sous windows mémoire vidéo) attention aux serveurs !!!

Exemples :

```
C:\OpenSSL\out32dll>openssl genrsa -aes256
Loading 'screen' into random state - done
Generating RSA private key, 512 bit long modulus
....++++++++++++
........++++++++++++
unable to write 'random state'
e is 65537 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,0D40E526A28261A5F1ECDA4CE7381067

SGyGshCsVYLP1jNOjsN84NfcyI1j1lVER4jiC917wXvj7VQ5t7MAi7QaOzl1YDVa
mkOG/kDgMDvE4+uOjwNmf04fanb+QYbP//PBWgGUUwomuZGCXzvkrjgT4Dd1Yun8
cAmD3++Y6zJyP3bp1B2zJwe1zkyzDqB6eMaP5WNGrcNzc7WP4IS9j6B6hUBcanJY
ISJ+uEnH5ddWSuznJJT6RA242URNIw3OjoUA4D+gDb1oqNakS3YU/YvAsJbx9416
JjUzVqqtxemslJvc+57v7p39neMYq9d7MHNBQYge5zW0cPxsClxOn5jwKvJ5xDWd
wBmDsjAozXPrWMf8xV2AwPfMn47NXFERVjtohd3Viy/REgV533hkXDujqct5m1On
aOnTLqfuiKvfOAJnW6kL9pcu7K4JOcv/VglDIpZy7q4=
-----END RSA PRIVATE KEY-----
C:\OpenSSLL\out32dll>
```

```
C:\OpenSSL\out32dll>echo test | openssl genrsa -aes256 -passout stdin
```

```
C:\OpenSSL\out32dll>openssl genrsa -aes256 -passout pass:test
```

```
C:\OpenSSL\out32dll>openssl genrsa -aes256 -passout file:motdepasse.txt
```

```
C:\OpenSSL\out32dll>set motdepasse=test
C:\OpenSSL\out32dll>openssl genrsa -aes256 -passout env:motdepasse
```

```
C:\OpenSSL\out32dll>openssl rsa -in pvk.pem -text
Enter pass phrase for pvk.pem:
Private-Key: (512 bit)
modulus:
    00:98:24:c6:b7:6d:f9:6f:64:02:ac:e7:f5:bd:ca:
    b8:28:d8:ec:10:77:5f:a1:fe:c3:40:91:aa:c4:d4:
    e8:3b:97:21:86:d6:84:19:77:57:21:37:26:68:7d:
    92:6a:ec:69:90:09:ad:d2:30:f5:a9:bd:18:4a:ef:
    b6:e3:7a:3b:bb
publicExponent: 65537 (0x10001)
privateExponent:
    00:84:5f:1a:f5:e5:1d:2c:a9:5b:1a:8d:06:e6:06:
```

```
    46:8c:63:8f:a0:13:fc:84:b9:5f:b0:02:0d:0e:0c:
    1f:b2:17:c8:8b:ae:cf:97:c2:c4:21:f1:eb:57:b2:
    68:81:6c:79:34:95:ac:89:b5:a6:fc:87:a4:ca:c1:
    2b:d3:b2:b1:a1
prime1:
    00:c6:39:14:dd:72:6d:db:0f:98:5d:0c:97:f8:35:
    f7:9f:98:d0:f0:dc:de:10:c5:f4:30:e1:07:c8:9d:
    07:1c:95
prime2:
    00:c4:7d:5e:1c:12:82:3d:23:7f:71:68:8c:c2:9d:
    4f:2e:47:1b:47:ec:bf:6a:34:68:3a:11:00:85:c0:
    8e:93:0f
exponent1:
    7c:76:4e:f1:94:ee:01:84:53:48:e9:a5:6a:46:88:
    8a:ff:6b:02:31:cc:85:7f:a6:0a:22:e1:be:47:4a:
    b5:d9
exponent2:
    00:85:fb:36:80:cd:de:20:f2:63:a4:61:36:1a:3e:
    4a:0f:b4:4c:d5:ee:89:57:52:05:00:13:37:d7:fc:
    45:3b:e1
coefficient:
    2f:fa:c5:44:fc:72:b1:df:58:ea:8d:ce:c8:b6:06:
    f5:2b:8b:72:79:5d:9a:b0:40:b8:00:b4:0a:9b:b8:
    78:c1
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIIBOwIBAAJBAJgkxrdt+W9kAqzn9b3KuCjY7BB3X6H+wOCRqsTU6DuXIYbWhBl3
VyE3Jmh9kmrsaZAJrdIw9am9GErvtuN6O7sCAwEAAQJBAIRfGvXlHSypWxqNBuYG
Roxjj6AT/IS5X7ACDQ4MH7IXyIuuz5fCxCHx61eyaIFseTSVrIm1pvyHpMrBK9Oy
saECIQDGORTdcm3bD5hdDJf4NfefmNDw3N4QxfQw4QfInQcclQIhAMR9XhwSgj0j
f3FojMKdTy5HG0fsv2o0aDoRAIXAjpMPAiB8dk7xlO4BhFNI6aVqRoiK/2sCMcyF
f6YKIuG+R0q12QIhAIX7NoDN3iDyY6RhNho+Sg+0TNXuiVdSBQATN9f8RTvhAiAv
+sVE/HKx31jqjc7Itgb1K4tyeV2asEC4ALQKm7h4wQ==
-----END RSA PRIVATE KEY-----
```

Generation de la CSR
OPENSSL REQ

Paramètres courants de la ligne de commande	
-inform [DER|PEM]	Format base64 ou binaire (ASN1)
-outform [DER|PEM]	Format base64 ou binaire
-in file	
-out file	
-text	Extraction au format texte de la CSR
-pubkey	Extraction de la clef publique de la CSR
-noout	Supprime la sortie de la CSR
-verify	Vérification de la signature de la CSR
-modulus	Extraction du modulo de la clef
-nodes	Pas de protection de la clef
-subject	Extraction des Distinguished Names
-key file	Précise la pvk à utiliser
-passin arg	Mot de passe de la clef privée
-keyform [DER|PEM|NETSCAPE|IISSGC|PKCS12]	Formats de la clef privée
-keyout file	Fichier de sortie de la clef privée si générée
-newkey rsa :[128|256|512]	Chiffrement de la clef privée si générée
-[md5|sha1|md2|mdc2|md4]	Algorithme de signature
-config file	Fichier de configuration openssl.cnf
-subj arg	Request subject
-new	Nouvelle requête
-batch	Pas de saisie
-x509	Extraction d'un certificat au lieu d'une CSR
-days	Durée de validité du certificat
-set_serial	Numéro de serie du certificat

Exemples :

CSR minimale avec génération de la PVK

```
C:\OpenSSL>openssl req -new -config openssl.cnf -passout pass:test
```

openssl.cnf

```
[ req ]
distinguished_name  = req_distinguished_name
default_bits        = 1024
default_keyfile     = privkey.pem
default_md          = md5

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = FR
countryName_min             = 2
countryName_max             = 2
countryName_value           = FR
```

Liste des Distinguished Names (\crypto\objects\ objects.txt):
commonName(CN),surname(SN),serialNumber,countryName(C), LocalityName(L),
stateOrProvinceName(ST), streetAddress, organizationName(O), organizationalUnitName(OU), title,

description, postalCode, name(name), givenName(GN), initials, generationQualifier, x500UniqueIdentifier, dnQualifier (dnQualifier), pseudonym, role

```
C:\OpenSSL>openssl req -new -config openssl.cnf -passout pass:test -text -key
pvk.pem
Enter pass phrase for pvk.pem:
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) []:FR
adresse []:rue saint sebastien
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=FR/streetAddress=rue saint sebastien
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:98:24:c6:b7:6d:f9:6f:64:02:ac:e7:f5:bd:ca:
                    b8:28:d8:ec:10:77:5f:a1:fe:c3:40:91:aa:c4:d4:
                    e8:3b:97:21:86:d6:84:19:77:57:21:37:26:68:7d:
                    92:6a:ec:69:90:09:ad:d2:30:f5:a9:bd:18:4a:ef:
                    b6:e3:7a:3b:bb
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: md5WithRSAEncryption
        26:a3:61:7a:c3:16:be:47:f8:8a:9e:f1:2e:a9:39:26:86:99:
        54:b7:7e:45:be:ba:c7:0e:25:0a:19:e0:47:aa:b6:dc:24:c3:
        34:f5:c0:74:1a:8f:09:fb:02:ba:70:7f:12:c0:be:64:86:7c:
        e8:88:e2:73:34:54:8e:43:33:42
-----BEGIN CERTIFICATE REQUEST-----
MIHlMIGQAgEAMCsxCzAJBgNVBAYTAkZSMRwwGgYDVQQJExNydWUgc2FpbnQgc2Vi
YXN0aWVuMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAJgkxrdt+w9kAqzn9b3KuCjY
7BB3X6H+w0CRqsTU6DuXIYbWhBl3VyE3Jmh9kmrsaZAJrdIw9am9GErvtuN6O7sC
AwEAAaAAMA0GCSqGSIb3DQEBBAUAA0EAJqNhesMWvkf4ip7xLqk5JoaZVLd+Rb66
xw4lChngR6q23CTDNPXAdBqPCfsCunB/EsC+ZIZ86IjiczRUjkMzQg==
-----END CERTIFICATE REQUEST-----
```

Génération en mode batch de la CSR avec fichier de configuration

```
C:\ OpenSSL >openssl req -config openssl.cnf -passout pass:test -key pvk.pem -new
-passin pass:test -batch -text -noout
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=FR
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:98:24:c6:b7:6d:f9:6f:64:02:ac:e7:f5:bd:ca:
                    b8:28:d8:ec:10:77:5f:a1:fe:c3:40:91:aa:c4:d4:
                    e8:3b:97:21:86:d6:84:19:77:57:21:37:26:68:7d:
                    92:6a:ec:69:90:09:ad:d2:30:f5:a9:bd:18:4a:ef:
                    b6:e3:7a:3b:bb
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: md5WithRSAEncryption
        79:d6:7d:af:79:a6:10:2b:33:45:89:8a:71:aa:9c:5c:89:7b:
        05:1f:c3:6a:f6:13:30:8e:d0:5e:69:c6:aa:0a:ea:0e:5c:eb:
        04:2b:c9:2e:db:37:ef:57:09:96:72:50:12:b8:fc:75:5f:de:
        bf:9b:4e:5e:fa:68:ac:0b:2d:d5
```

avec

openssl.cnf

```
...
 [ req_distinguished_name ]
countryName                = Country Name (2 letter code)
countryName_default        = FR
...
```

Génération en mode batch de la CSR sans fichier de configuration

```
C:\OpenSSL>openssl req -config openssl.cnf -passout pass:te
st -key pvk.pem -new   -passin pass:test -batch -subject -subj /L=LILLE/CN=TEST
-noout
subject=/L=LILLE/CN=TEST
```

la section [req_distinguished_name] peut être vide

vérification de la PVK et de la CSR

```
C:\OpenSSL>openssl req -verify -in csr.pem -key pvk2.pem -passin pass:test -config
openssl.cnf -noout
verify failure
3900:error:04077077:rsa routines:RSA_verify:wrong signature length:.\crypto\rsa\
rsa_sign.c:154:
3900:error:0D0C5006:asn1 encoding routines:ASN1_item_verify:EVP lib:.\crypto\asn
1\a_verify.c:168:
```

Attributs étendus de la CSR

```
C:\OpenSSL>openssl req -config openssl.cnf -passout pass:test -key pvk.pem -new
-passin pass:test  -text -batch -noout
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=FR
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:98:24:c6:b7:6d:f9:6f:64:02:ac:e7:f5:bd:ca:
                    b8:28:d8:ec:10:77:5f:a1:fe:c3:40:91:aa:c4:d4:
                    e8:3b:97:21:86:d6:84:19:77:57:21:37:26:68:7d:
                    92:6a:ec:69:90:09:ad:d2:30:f5:a9:bd:18:4a:ef:
                    b6:e3:7a:3b:bb
                Exponent: 65537 (0x10001)
        Attributes:
            countryName               :FR
    Signature Algorithm: md5WithRSAEncryption
        06:d3:bc:f8:f1:75:c5:3d:d0:bf:b4:af:24:21:6f:65:b0:ea:
        9f:26:6d:86:2e:89:23:0d:ff:8a:8e:19:98:7d:25:13:02:61:
        ab:82:66:a4:f2:f8:32:23:aa:da:a3:6a:07:36:fd:0a:c3:7b:
        5f:b5:fe:e4:fe:1c:5c:f2:99:d0
```

openssl.cnf

```
[ req ]
attributes          = req_attributes
...
[ req_attributes ]
countryName                = Country Name (2 letter code)
countryName_default        = FR
...
```

Génération d'une nouvelle autorité

```
C:\OpenSSL>openssl req -in csr.pem -x509 -config openssl.cnf -key pvk.pem -text
Enter pass phrase for pvk.pem:
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
```

```
            9e:10:61:50:10:18:c0:94
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=FR, ST=FRANCE, L=MARCQ EN BAROEUL, OU=TEST, CN=Arnaud Grandvil
le/emailAddress=contact@grandville.net
        Validity
            Not Before: Dec 13 13:27:45 2009 GMT
            Not After : Jan 12 13:27:45 2010 GMT
        Subject: C=FR, ST=FRANCE, L=MARCQ EN BAROEUL, OU=TEST, CN=Arnaud Grandvi
lle/emailAddress=contact@grandville.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:e5:c9:0a:dc:06:e4:56:df:19:29:1f:eb:58:85:
                    e0:89:01:23:f8:4d:2a:6b:68:b9:fc:43:a8:43:93:
                    cd:90:f5:8a:3a:32:a2:d2:2a:c6:f6:83:8f:35:0f:
                    16:9e:81:05:d3:9b:68:71:8e:d4:cb:8d:a5:3f:33:
                    bb:72:90:c8:51
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        0e:01:90:a8:87:15:76:5b:e6:9d:5d:8c:dc:a0:f2:60:3b:b1:
        70:61:88:b5:a5:40:31:dd:91:c5:77:de:35:29:20:d6:bf:de:
        84:58:31:5d:6b:9b:ef:da:b8:28:ae:ea:e8:6b:74:da:6a:36:
        dd:84:54:aa:99:1b:46:af:12:cc
```

openssl.cnf

```
[ req ]
x509_extensions     = v3_ca
...

[ v3_ca ]
basicConstraints=CA:TRUE,pathlen:1
keyUsage = keyCertSign
```

Création du certificat
OPENSSL X509

Paramètres courants de la ligne de commande	
-inform [DER|PEM]	Format base64 ou binaire (ASN1)
-outform [DER|PEM]	Format base64 ou binaire
-keyform	Format de la clef privée
-CAform arg	Format de la CA
-CAkeyform arg	Format de la clef de la CA
-in file	
-out file	
-passin arg	Mot de passe de la clef privé de la CA
-noout	Supprime la sortie du certificat
-days arg	Durée de validité du certificat
-req	Le fichier en entrée est une CSR
-set_serial arg	Numéro de série du certificat au format decimal
-text	Extraction au format texte de la CSR

Exemples

```
C:\OpenSSL>openssl x509 -in csr.pem -req -CA ..\config\CA.crt  -CAkey
..\config\CA_pvk.pem -passin pass:test -set_serial 9999 -text
Loading 'screen' into random state - done
Signature ok
subject=/C=FR/streetAddress=rue saint sebastien
Getting CA Private Key
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 9999 (0x270f)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=FR, ST=FRANCE, L=LILLE, O=GRANDVILLE/emailAddress=contact@gran
dville.net
        Validity
            Not Before: Dec 11 13:09:17 2009 GMT
            Not After : Jan 10 13:09:17 2010 GMT
        Subject: C=FR/streetAddress=rue saint sebastien
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:98:24:c6:b7:6d:f9:6f:64:02:ac:e7:f5:bd:ca:
                    b8:28:d8:ec:10:77:5f:a1:fe:c3:40:91:aa:c4:d4:
                    e8:3b:97:21:86:d6:84:19:77:57:21:37:26:68:7d:
                    92:6a:ec:69:90:09:ad:d2:30:f5:a9:bd:18:4a:ef:
                    b6:e3:7a:3b:bb
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        a1:f2:72:ea:80:18:6a:3e:6c:ef:28:72:ce:5d:d0:ee:3d:58:
        95:2d:26:d9:03:20:1e:2b:96:69:bb:5c:fb:04:31:d7:22:52:
        e6:c1:d6:23:64:58:c7:5b:03:61:bb:21:18:e6:e9:80:02:ad:
        14:7f:da:a1:de:fb:39:25:a2:12:8d:71:7f:12:c7:2c:f2:5c:
        6e:56:cf:cb:33:76:42:ee:9e:71:98:05:fa:a8:e0:87:31:c1:
        aa:8f:2d:3c:1b:ad:3d:cb:71:00:2c:40:e8:68:f1:14:6d:ee:
        f3:07:16:a8:69:95:99:f7:d8:1f:cd:50:96:b6:bf:e3:b5:51:
        53:a1
-----BEGIN CERTIFICATE-----
MIIBwTCCASoCAicPMA0GCSqGSIb3DQEBBQUAMGoxCzAJBgNVBAYTAkZSMQ8wDQYD
VQQIEwZGUkFOQ0UxDjAMBgNVBAcTBUxJTExFMRMwEQYDVQQKEwpHUkFORFZJTExF
MSUwIwYJKoZIhvcNAQkBFhZjb250YWN0QGdyYW5kdmlsbGUubmV0MB4XDTA5MTIx
MTEzMDkxN1oXDTEwMDExMDEzMDkxN1owKzELMAkGA1UEBhMCRlIxHDAaBgNVBAkT
E3J1ZSBzYWludCBzZWJhc3RpZW4wXDANBgkqhkiG9w0BAQEFAANLADBIAkEAmCTG
t235b2QCrOf1vcq4KNjsEHdfof7DQJGqxNToO5chhtaEGXdXITcmaH2SauxpkAmt
```

```
0jD1qb0YSu+243o7uwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAKHycuqAGGo+bO8o
cs5dOO49WJUtJtkDIB4rlmm7XPsEMdciUubB1iNkWMdbA2G7IRjm6YACrRR/2qHe
+zklohKNcX8SxyzyXG5Wz8szdkLunnGYBfqo4IcxwaqPLTwbrT3LcQAsQOho8RRt
7vMHFqhplZn32B/NUJa2v+O1UVOh
-----END CERTIFICATE-----
unable to write 'random state'
```

Certificat avec extensions x509v3

ext.cnf

```
basicConstraints = critical,CA:true
keyUsage= digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment,
keyAgreement, keyCertSign, cRLSign, encipherOnly, decipherOnly
```

```
C:\OpenSSL>openssl x509 -in csr.pem -req -CA ..\config\CA.crt  -CAkey
..\config\CA_pvk.pem -passin pass:test -set_serial 9999 -text -extfile ext.cnf
Loading 'screen' into random state - done
Signature ok
subject=/C=FR/streetAddress=rue saint sebastien
Getting CA Private Key
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 9999 (0x270f)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=FR, ST=FRANCE, L=LILLE, O=GRANDVILLE/emailAddress=contact@gran
dville.net
        Validity
            Not Before: Dec 11 13:19:21 2009 GMT
            Not After : Jan 10 13:19:21 2010 GMT
        Subject: C=FR/streetAddress=rue saint sebastien
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:98:24:c6:b7:6d:f9:6f:64:02:ac:e7:f5:bd:ca:
                    b8:28:d8:ec:10:77:5f:a1:fe:c3:40:91:aa:c4:d4:
                    e8:3b:97:21:86:d6:84:19:77:57:21:37:26:68:7d:
                    92:6a:ec:69:90:09:ad:d2:30:f5:a9:bd:18:4a:ef:
                    b6:e3:7a:3b:bb
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha1WithRSAEncryption
        5b:61:72:83:19:bf:47:74:14:9e:3a:9b:fb:f7:be:52:f8:53:
        95:fb:c4:cd:e4:f6:25:f8:4e:32:f3:a3:f8:4b:7e:4b:ff:02:
        d6:f5:66:6b:a2:5d:b0:a5:93:37:b6:98:33:bb:d3:0d:5f:0b:
        7b:66:4c:ef:fb:b3:30:80:d7:63:38:c9:28:cb:d5:ea:fe:16:
        0d:ea:6b:d2:93:f5:8d:a4:b8:40:5d:73:83:93:d0:48:87:fd:
        5c:7d:c9:b0:d9:5f:0f:59:21:22:c0:ee:83:f0:2c:b0:b6:5f:
        e9:a8:9b:20:b1:7f:c8:a7:d7:0c:48:0c:b3:14:8a:90:24:70:
        9e:a1
-----BEGIN CERTIFICATE-----
MIIB2DCCAUGgAwIBAgICJw8wDQYJKoZIhvcNAQEFBQAwajELMAkGA1UEBhMCRlIx
DzANBgNVBAgTBkZSQU5DRTEOMAwGA1UEBxMFTElMTEUxEzARBgNVBAoTCkdSQU5E
VklMTEUxJTAjBgkqhkiG9w0BCQEWFmNvbnRhY3RAZ3JhbmR2aWxsZS5uZXQwHhcN
MDkxMjExMTMxOTIxWhcNMTAwMTEwMTMxOTIxWjArMQswCQYDVQQGEwJGUjEcMBoG
A1UECRMTcnVlIHNhaW50IHNlYmFzdGllbjBcMA0GCSqGSIb3DQEBAQUAA0sAMEgC
QQCYJMa3bflvZAKs5/W9yrgo2OwQd1+h/sNAkarE1Og7lyGG1oQZd1chNyZofZJq
7GmQCa3SMPWpvRhK77bjeju7AgMBAAGjEDAOMAwGA1UdEwQFMAMBAf8wDQYJKoZI
hvcNAQEFBQADgYEAW2Fygxm/R3QUnjqb+/e+UvhTlfvEzeT2JfhOMvOj+Et+S/8C
1vVma6JdsKWTN7aYM7vTDV8Le2ZM7/uzMIDXYzjJKMvV6v4WDepr0pP1jaS4QF1z
g5PQSIf9XH3JsNlfD1khIsDug/AssLZf6aibILF/yKfXDEgMsxSKkCRwnqE=
-----END CERTIFICATE-----
unable to write 'random state'
```

Création du certificat
OPENSSL CA

Paramètres courants de la ligne de commande	
-inform [DER|PEM]	Format base64 ou binaire (ASN1)
-outform [DER|PEM]	Format base64 ou binaire
-keyform	Format de la clef privée
-CAform arg	Format de la CA

ca.cnf

```
dir          = ./demoCA          # Where everything is kept
RANDFILE     = $dir/.rand # private random number file

[ ca ]
default_ca   = CA_default

[ CA_default ]
database     = $dir/index.txt    # database index file.
new_certs_dir= $dir/newcerts     # default place for new certs.
certificate  = $dir/cacert.pem   # The CA certificate
serial       = $dir/serial.txt   # The current serial number
private_key  = $dir/cakey.pem    # The private key
RANDFILE     = $dir/.rand        # private random number file

default_days = 365               # how long to certify for
default_md   = sha1              # which md to use.
policy       = policy_match

[ policy_match ]
commonName          = supplied # optional, match
```

structure de répertoire

```
/demoCA
              /newcerts
               AAAA.pem
        AAAB.pem
                ....
  cacert.pem
  cakey.pem
  index.txt
  index.txt.attr
  serial.txt
  .rnd
```

index.txt

```
V      101213101127Z       AAC7   unknown      /C=FR/CN=Arnaud Grandville/L=MARCQ EN
BAROEUL/emailAddress=contact@grandville.net
```

V :valide

E :expiré

R :révoqué

format :

Etat ;Expiration date ;Revocation date ;serial ;fichier ;...

```
C:\OpenSSL>openssl ca -config ca.cnf -in csr.pem
```

```
Using configuration from ca.cnf
Loading 'screen' into random state - done
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'FRANCE'
localityName          :PRINTABLE:'MARCQ EN BAROEUL'
organizationalUnitName:PRINTABLE:'TEST'
commonName            :PRINTABLE:'Arnaud Grandville'
emailAddress          :IA5STRING:'contact@grandville.net'
Certificate is to be certified until Dec 13 10:07:38 2010 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 43718 (0xaac6)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=AU, ST=QLD, O=Mincom Pty. Ltd., OU=CS, CN=SSLeay demo server
        Validity
            Not Before: Dec 13 10:07:38 2009 GMT
            Not After : Dec 13 10:07:38 2010 GMT
        Subject: C=FR, CN=Arnaud Grandville, L=MARCQ EN BAROEUL/emailAddress=con
tact@grandville.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:e5:c9:0a:dc:06:e4:56:df:19:29:1f:eb:58:85:
                    e0:89:01:23:f8:4d:2a:6b:68:b9:fc:43:a8:43:93:
                    cd:90:f5:8a:3a:32:a2:d2:2a:c6:f6:83:8f:35:0f:
                    16:9e:81:05:d3:9b:68:71:8e:d4:cb:8d:a5:3f:33:
                    bb:72:90:c8:51
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        61:2d:ff:45:49:11:df:02:45:4c:fb:69:29:69:68:18:72:fb:
        6d:59:03:29:47:e8:66:d9:22:0f:8e:e7:18:15:a3:bb:c4:26:
        fe:35:60:0f:65:d9:b0:2c:60:44:7d:9a:71:55:33:7d:cf:63:
        c5:b0:dd:9a:ac:10:37:f3:de:cb
-----BEGIN CERTIFICATE-----
MIIBtzCCAWECAwCqxjANBgkqhkiG9w0BAQUFADBgMQswCQYDVQQGEwJBVTEMMAoG
A1UECBMDUUxEMRkwFwYDVQQKExBNaW5jb20gUHR5LiBMdGQuMQswCQYDVQQLEwJD
UzEbMBkGA1UEAxMSU1NMZWF5IGRlbW8gc2VydmVyMB4XDTA5MTIxMzEwMDczOFoX
DTEwMTIxMzEwMDczOFowazELMAkGA1UEBhMCRlIxGjAYBgNVBAMTEUFybmF1ZCBH
cmFuZHZpbGxlMRkwFwYDVQQHExBNQVJDUSBFTiBCQVJPUVVMMSUwIwYJKoZIhvcN
AQkBFhZjb250YWN0QGdyYW5kdmlsbGUubmV0MFwwDQYJKoZIhvcNAQEBBQADSwAw
SAJBAOXJCtwG5FbfGSkf61iF4IkBI/hNKmtoufxDqEOTzZD1ijoyotIqxvaDjzUP
Fp6BBdObaHGO1MuNpT8zu3KQyFECAwEAATANBgkqhkiG9w0BAQUFAANBAGEt/0VJ
Ed8CRUz7aSlpaBhy+21ZAylH6GbZIg+O5xgVo7vEJv41YA9l2bAsYER9mnFVM33P
Y8Ww3ZqsEDfz3ss=
-----END CERTIFICATE-----
Data Base Updated
```

extensions de certificats

ca.cnf

```
[ CA_default ]
...
x509_extensions     = usr_cert          # The extentions to add to the cert

[ usr_cert ]
basicConstraints=critical,CA:FALSE
keyUsage =
digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyAgreement,keyCe
rtSign,cRLSign,encipherOnly,decipherOnly
extendedKeyUsage= serverAuth,clientAuth,emailProtection,codeSigning
crlDistributionPoints=URI:http://www.domain1.dom/ca-
crl.pem,URI:http://www.domain2.dom/ca-crl.pem
subjectAltName=DNS:www.test.com,URI:www.dom.org,email:move,IP:fe80::1862:1316:3f57:
fefe
```

```
certificatePolicies=2.16.840.1.113733.1.7.23.6
```

le certificat EBAY

```
C:\OpenSSL>openssl x509 -in ebay.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            3b:32:69:69:32:58:1e:33:d0:ee:e5:2a:19:70:27:75
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=VeriSign, Inc., OU=VeriSign Trust Network, OU=Terms of u
se at https://www.verisign.com/rpa (c)06, CN=VeriSign Class 3 Extended Validatio
n SSL CA
        Validity
            Not Before: Jan  8 00:00:00 2009 GMT
            Not After : Jan 24 23:59:59 2011 GMT
        Subject: 1.3.6.1.4.1.311.60.2.1.3=US/1.3.6.1.4.1.311.60.2.1.2=Delaware/2
.5.4.15=V1.0, Clause 5.(b)/serialNumber=2871352, C=US/postalCode=95125, ST=Calif
ornia, L=San Jose/streetAddress=2145 Hamilton Ave, O=eBay Inc., OU=Site Operatio
ns European, CN=signin.ebay.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:df:6b:e0:d2:8c:3d:d3:0c:47:c3:c6:63:69:3d:
                    c6:b9:ca:9f:be:58:85:2a:21:b9:2e:52:62:03:b5:
                    3c:8d:f8:ab:bb:4b:d6:d9:e5:6a:be:d6:d0:5e:56:
                    d9:ed:56:3b:05:8c:a3:11:e7:49:34:28:e9:32:ce:
                    b2:1e:98:e4:ef:6c:e7:8e:02:eb:b3:ba:9b:f3:97:
                    5e:b7:1d:77:c3:b2:0b:1b:3d:52:97:9a:9d:07:53:
                    03:6f:d4:4d:99:39:09:b0:72:21:c9:a2:62:21:b9:
                    14:dd:2f:df:74:35:7d:fc:96:f5:13:4e:1a:eb:0e:
                    e0:37:30:d3:23:e0:22:57:03:93:76:77:25:45:c1:
                    23:29:25:e1:68:b7:fb:c3:82:65:b3:6b:14:f3:03:
                    39:c8:b2:0f:35:02:3f:6b:c5:6b:ef:c9:7d:36:6e:
                    24:53:27:b4:53:92:b2:d2:3b:c1:88:0c:14:ca:e9:
                    cf:45:71:0f:fa:c3:df:dd:a5:09:b2:4e:cc:5d:3d:
                    96:cf:3b:d8:00:ba:b4:c5:59:a4:d6:80:82:d9:92:
                    6a:90:58:d7:4e:ce:ea:9d:5b:75:5a:64:9b:ba:d2:
                    2b:26:3f:12:07:a8:fb:f4:5e:3a:88:c3:8f:bf:11:
                    ec:a2:40:e7:cf:89:8e:ba:61:3e:9a:c5:2c:9a:e6:
                    88:43
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:signin.ebay.com, DNS:signin.ebay.at, DNS:signin.ebay.be, DNS
:signin.ebay.ch, DNS:signin.ebay.de, DNS:signin.ebay.es, DNS:signin.ebay.fr, DNS
:signin.ebay.ie, DNS:signin.ebay.it, DNS:signin.ebay.nl, DNS:signin.ebay.pl, DNS
:signin.befr.ebay.be, DNS:signin.benl.ebay.be, DNS:signin.ebay.co.uk
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                13:2F:BD:CD:A5:DC:3F:30:EA:F2:6E:CC:11:68:53:B5:3D:2C:11:F8
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 CRL Distribution Points:
                URI:http://EVSecure-crl.verisign.com/EVSecure2006.crl
            X509v3 Certificate Policies:
                Policy: 2.16.840.1.113733.1.7.23.6
                  CPS: https://www.verisign.com/rpa
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Authority Key Identifier:
                keyid:FC:8A:50:BA:9E:B9:25:5A:7B:55:85:4F:95:00:63:8F:E9:58:6B:43
            Authority Information Access:
                OCSP - URI:http://EVSecure-ocsp.verisign.com
                CA Issuers - URI:http://EVSecure-aia.verisign.com/EVSecure2006.cer
            1.3.6.1.5.5.7.1.12:
                0`.^.\0Z0X0V..image/gif0!0.0...+......Kk.(.....R8.).K..!..0&.$ht
tp://logo.verisign.com/vslogo1.gif
    Signature Algorithm: sha1WithRSAEncryption
        5d:ee:49:24:ae:92:d7:ca:d6:a1:31:12:e6:e9:4c:fb:46:e8:
```

```
        59:81:64:08:df:89:f5:cf:f2:b8:88:68:15:e5:c4:dd:3b:bf:
        3f:ff:cf:3f:e3:d5:99:6f:6c:0b:f1:90:d5:f1:3e:00:d9:bd:
        da:be:37:69:9c:55:a1:44:bb:f8:d7:a7:b0:0c:0d:04:f6:78:
        47:dc:28:67:8a:4d:0b:80:68:b9:27:8a:62:3e:6b:db:45:c1:
        cb:fc:84:8c:e3:ba:52:6d:3a:d3:dc:04:75:1f:e6:fa:02:80:
        54:37:21:71:08:e3:f8:4c:93:7c:65:a5:8a:34:5c:a7:e4:d5:
        9a:45:d2:06:34:fc:9d:31:15:d1:78:0b:6a:59:35:63:c7:aa:
        4c:82:22:a3:7e:d5:f2:6a:d8:22:eb:a7:a0:fb:33:a2:16:be:
        1a:7d:ac:af:54:9e:c1:cb:04:34:b2:c0:f7:b8:18:07:16:da:
        f3:d5:9a:b9:d1:69:56:f0:50:56:dd:3a:e9:a8:7a:50:26:80:
        ad:77:7c:8c:25:52:28:fe:3f:2c:2e:11:6d:7d:15:f0:83:78:
        81:43:0b:fa:e5:6a:f9:3a:d2:02:ec:4d:e3:52:b6:3c:a5:8d:
        e8:77:08:1e:8e:3f:7b:a1:cb:25:f6:ec:60:76:07:b5:b7:46:
        d7:ff:3a:7e
```

Le certificat www.fnac.com

```
C:\OpenSSL>openssl x509 -in fnac.com.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            74:a3:fa:be:ac:07:df:e1:0e:9d:7a:d4:b5:07:46:bb
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=VeriSign, Inc., OU=VeriSign Trust Network, OU=Terms of u
se at https://www.verisign.com/rpa (c)06, CN=VeriSign Class 3 Extended Validatio
n SSL SGC CA
        Validity
            Not Before: Aug 31 00:00:00 2009 GMT
            Not After : Aug 31 23:59:59 2010 GMT
        Subject: 1.3.6.1.4.1.311.60.2.1.3=FR/2.5.4.15=V1.0, Clause 5.(b)/serialN
umber=775661390, C=FR/postalCode=94200, ST=Hauts de Seine, L=IVRY SUR SEINE/stre
etAddress=9, Rue des Bateaux Lavoirs, O=FNAC SA, OU=FNAC Direct, CN=www.fnac.com

        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:cd:2c:c5:95:69:10:82:6c:41:ad:d7:37:03:cd:
                    9d:95:37:58:b9:38:0e:49:73:21:52:3d:6e:a3:31:
                    f1:95:fa:55:83:89:f9:0c:2f:b5:77:ad:a1:4b:1b:
                    68:81:96:15:1e:f5:a0:97:95:53:43:50:0f:fb:79:
                    7b:4c:2c:95:51:37:e8:78:1b:d7:49:a1:f3:47:b2:
                    a9:77:d0:8b:af:23:92:85:3b:38:a3:73:d6:a5:b6:
                    60:0f:0e:03:75:cd:87:f5:91:51:ed:8d:48:5c:f7:
                    99:75:9d:5b:ce:ce:27:67:14:4a:a7:41:21:35:b4:
                    4a:8d:ed:66:90:85:0c:e3:9b:21:71:e4:24:ba:a5:
                    01:29:59:fc:bd:ba:b3:ce:1a:f8:aa:ad:c8:71:23:
                    13:d6:91:83:12:42:65:31:16:d3:28:99:39:1f:30:
                    db:b1:67:d6:2d:3b:04:96:65:8e:37:5f:26:e1:e4:
                    e6:18:99:7d:09:7b:3d:9a:f4:b0:ed:49:06:fa:7e:
                    4f:68:cf:db:5f:72:e1:81:b9:07:fd:9b:fd:2c:02:
                    65:08:f5:60:6f:f3:2a:76:6f:d9:68:7c:6f:bd:60:
                    bd:d3:0b:d5:5a:2b:33:6e:fa:f5:b7:80:f0:43:96:
                    84:6d:41:93:1f:6f:fd:5a:91:16:31:76:0e:50:d8:
                    e8:b5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                D8:77:66:D7:9E:E3:C1:2F:41:9F:5C:5A:03:56:F6:67:28:1F:E3:4A
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Certificate Policies:
                Policy: 2.16.840.1.113733.1.7.23.6
                  CPS: https://www.verisign.com/rpa

            X509v3 CRL Distribution Points:
                URI:http://EVIntl-crl.verisign.com/EVIntl2006.crl

            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication, Ne
tscape Server Gated Crypto
```

```
            X509v3 Authority Key Identifier:
                keyid:4E:43:C8:1D:76:EF:37:53:7A:4F:F2:58:6F:94:F3:38:E2:D5:BD:DF

            Authority Information Access:
                OCSP - URI:http://EVIntl-ocsp.verisign.com
                CA Issuers - URI:http://EVIntl-aia.verisign.com/EVIntl2006.cer

            1.3.6.1.5.5.7.1.12:
                0`.^.\0Z0X0V..image/gif0!0.0...+......Kk.(.....R8.).K..!..0&.$ht
tp://logo.verisign.com/vslogo1.gif
    Signature Algorithm: sha1WithRSAEncryption
        11:55:c5:0a:c9:74:d2:54:62:46:78:00:2b:f2:0a:d2:e1:6d:
        ac:3d:46:9f:53:db:dc:84:22:fd:90:91:03:38:bb:4d:0f:5c:
        55:c8:91:0f:3a:50:8a:f0:67:fd:41:a2:0b:41:7f:fb:f0:6d:
        02:ee:73:68:a2:7c:d5:91:54:5f:fe:e7:54:c2:2d:9d:09:9a:
        e0:da:57:16:4f:37:18:1b:85:bc:0b:d7:64:71:59:b0:02:a2:
        b5:09:53:78:27:77:2b:2c:ff:82:77:6c:4e:e6:15:31:d8:b0:
        6b:9a:c3:2f:af:76:c1:c8:7f:47:f5:2d:82:89:89:c8:74:17:
        fc:62:15:97:96:1d:d5:3c:bf:e0:95:e8:10:6c:1b:ff:9a:4a:
        81:f6:7c:ab:c7:7b:b7:5f:0f:c9:4f:95:bc:7d:8b:a0:20:a8:
        82:fd:76:bf:3c:63:e9:00:9d:04:a8:25:af:05:de:c2:13:bf:
        f5:7e:c7:f0:c8:21:86:6b:bb:8c:7d:a0:5e:18:97:89:83:1f:
        f2:b1:a4:e1:98:e2:75:01:93:bd:eb:8f:ee:23:37:f5:5c:e0:
        86:1c:dc:05:38:75:e7:c0:b2:e8:7a:39:bd:33:da:92:a0:44:
        1c:e0:ef:47:2f:08:e8:a0:50:fe:8f:a9:51:3e:fb:5e:eb:3f:
        c5:b7:4b:96
```

certificat SSL Extended Validation, X509v3 Certificate Policies

Révocation du certificat
OPENSSL CA

```
C:\OpenSSL>openssl ca -config ca.cnf -revoke demoCA\newcerts\AAC8.pem
Using configuration from ca.cnf
Loading 'screen' into random state - done
Revoking Certificate AAC8.
Data Base Updated
```

le fichier index.txt conserve une trace du certificat révoqué

index.txt

```
V       101213101127Z               AAC7    unknown       /C=FR/CN=Arnaud Grandville
R       101213104424Z091213104747ZAAC8    unknown       /C=FR/CN=Arnaud Grandville
```

Génération de la CRL pour diffusion

ca.cnf

```
[ CA_default ]
crl_dir             = $dir/crl          # Where the issued crl are kept
crl                 = $dir/crl.pem            # The current CRL
crlnumber           = $dir/crlserial.txt      # the current crl number
default_crl_days    = 30
```

```
C:\OpenSSL>openssl ca -config ca.cnf -gencrl
Using configuration from ca.cnf
Loading 'screen' into random state - done
-----BEGIN X509 CRL-----
MIIBETCBvAIBATANBgkqhkiG9w0BAQUFADBgMQswCQYDVQQGEwJBVTEMMAoGA1UE
CBMDUUxEMRkwFwYDVQQKExBNaW5jb20gUHR5LiBMdGQuMQswCQYDVQQLEwJDUzEb
MBkGA1UEAxMSU1NMZWF5IGRlbW8gc2VydmVyFw0wOTEyMTMxMDUyNDBaFw0xMDAx
MTIxMDUyNDBaMBYwFAIDAKrIFw0wOTEyMTMxMDQ3NDdaoBAwDjAMBgNVHRQEBQID
AKrJMA0GCSqGSIb3DQEBBQUAA0EAGvgrj1rMkcWzlROmOyHErqMckI210eXReIyP
FvfbiKWY+SsUQedIXE78vCT70AoWXIrp7CBpsOxUSusOrhXKrQ==
-----END X509 CRL-----
```

Contenu d'un fichier CRL

```
C:\OpenSSL>openssl crl  -in crl.pem -text -noout
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /C=AU/ST=QLD/O=Mincom Pty. Ltd./OU=CS/CN=SSLeay demo server
        Last Update: Dec 13 10:53:54 2009 GMT
        Next Update: Jan 12 10:53:54 2010 GMT
        CRL extensions:
            X509v3 CRL Number:
                43722
Revoked Certificates:
    Serial Number: AAC8
        Revocation Date: Dec 13 10:47:47 2009 GMT
    Signature Algorithm: sha1WithRSAEncryption
        08:cb:c4:98:cc:22:45:c3:f2:4e:f2:e3:5b:4e:3c:3b:70:9f:
        3c:31:b7:cd:5a:08:e0:c1:ab:7e:10:fa:d0:59:db:da:78:6a:
        8a:34:3d:d8:dc:ca:c6:b2:4e:a6:78:c4:72:64:34:9f:9b:2f:
        53:eb:9e:dc:34:f4:73:b3:11:c2
```

openssl.cnf

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME            = .
RANDFILE        = $ENV::HOME/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file       = $ENV::HOME/.oid
oid_section     = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions    =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca    = CA_default        # The default ca section

####################################################################
[ CA_default ]

dir           = ./demoCA          # Where everything is kept
certs         = $dir/certs        # Where the issued certs are kept
crl_dir             = $dir/crl          # Where the issued crl are kept
database      = $dir/index.txt    # database index file.
#unique_subject     = no                # Set to 'no' to allow creation of
                                 # several ctificates with same subject.
new_certs_dir= $dir/newcerts            # default place for new certs.

certificate   = $dir/cacert.pem   # The CA certificate
serial        = $dir/serial             # The current serial number
crlnumber     = $dir/crlnumber    # the current crl number
                                 # must be commented out to leave a V1 CRL
crl           = $dir/crl.pem            # The current CRL
private_key   = $dir/private/cakey.pem# The private key
RANDFILE      = $dir/private/.rand      # private random number file

x509_extensions     = usr_cert          # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt      = ca_default        # Subject Name options
cert_opt      = ca_default        # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
```

```
# crl_extensions    = crl_ext

default_days = 365              # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md   = sha1             # which md to use.
preserve     = no               # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy       = policy_match

# For the CA policy
[ policy_match ]
countryName         = match
stateOrProvinceName = match
organizationName    = match
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName         = optional
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName    = optional
commonName          = supplied
emailAddress        = optional

####################################################################
[ req ]
default_bits        = 1024
default_keyfile     = privkey.pem
distinguished_name  = req_distinguished_name
attributes          = req_attributes
x509_extensions     = v3_ca      # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix  : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = AU
countryName_min                     = 2
countryName_max                     = 2

stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default       = Some-State

localityName                = Locality Name (eg, city)
```

```
0.organizationName         = Organization Name (eg, company)
0.organizationName_default= Internet Widgits Pty Ltd

# we can do this but it is not needed normally :-)
#1.organizationName       = Second Organization Name (eg, company)
#1.organizationName_default      = World Wide Web Pty Ltd

organizationalUnitName           = Organizational Unit Name (eg, section)
#organizationalUnitName_default  =

commonName                 = Common Name (eg, YOUR name)
commonName_max                   = 64

emailAddress               = Email Address
emailAddress_max           = 64

# SET-ex3                  = SET extension number 3

[ req_attributes ]
challengePassword          = A challenge password
challengePassword_min            = 4
challengePassword_max            = 20

unstructuredName           = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType               = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                  = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy
```

```
#nsCaRevocationUrl         = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always

[ proxy_cert_ext ]
# These extensions should be added when creating a proxy certificate

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE
```

```
# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType              = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                 = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl        = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

# This really needs to be in place for it to be a proxy certificate.
proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo
```

Format XML

Extensible Markup Language, Recommandation du W3C (XML 1.0 , 10 février 1998 et XML1.1 Février 2004) :

1. XML devrait pouvoir être utilisé sans difficulté sur Internet ;
2. XML devrait soutenir une grande variété d'applications ;
3. XML devra être compatible avec SGML ;
4. Il devrait être facile d'écrire des programmes traitant les documents XML ;
5. Le nombre d'options dans XML doit être réduit au minimum, idéalement à aucune ;
6. Les documents XML devraient être lisibles par l'homme et raisonnablement clairs ;
7. La conception de XML devrait être préparée rapidement ;
8. La conception de XML sera formelle et concise ;
9. Il devrait être facile de créer des documents XML ;
10. La concision dans le balisage de XML est de peu d'importance.

Déclaration, balises , attributs, commentaires et champs CDATA.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Cartographie -->
<Application Deployed="True" Name="OPE09021100" tri="OPE" ver="09021100"
Cluster="CLASXNPT05000002" slot="05" Timestamp="1">
   <Patch Name="LOG.PATCH010" date="Fri Oct 23 15:39:57 DFT 2009"
Timestamp="1" />
   <EAR Name="OPE-9.2.11.0" tri="OPE" ver="09021100" Timestamp="1" />
   <Lib Name="socle-im4.20.9.1" Timestamp="1" />
   <Lib Name="socle-architecture-pdt1.2.3" Timestamp="1" />
   <Lib Name="socle-applicatif-pdt-operation2.0.4.0" Timestamp="1" />
   <Lib Name="socle-objetscommuns-pdt1.6.1.1" Timestamp="1" />
   <Lib Name="stub6.10.0" Timestamp="1" />
   <SQL Name="OPE_code_decode_data" Date="20090916-163127"
Version="1.6.1.4" Timestamp="1" />
   <SQL Name="OPE_CBO_data" Date="20090916-163133" Version="4.20.8.0"
Timestamp="1" />
   <HTML>
     <![CDATA[<body>
serveur
</body>]]>
   </HTML>
</Application>
```

Les espaces de noms (1999)

```
<p xml:lang="fr">Cachez ce sein que je ne saurais voir.</p>
<p xml:lang="fr-FR">Ce programme a une bogue.</p>
<p xml:lang="fr-CA">Ce programme a un bogue.</p>
<sp qui="Faust" desc='leise' xml:lang="de">
   <l>Habe nun, ach! Philosophie,</l>
   <l>Juristerei, und Medizin</l>
   <l>und leider auch Theologie</l>
   <l>durchaus studiert mit hei&#223;em Bem&#252;h'n.</l>
</sp>
```

XSD

XML Schema Document est une norme XML (2001) qui est utilisée pour décrire une structure de documents XML
http://www.w3.org/XML/Schema
Editeur freeware Liquid XML Studio - http://www.liquid-technologies.com

```
<?xml version="1.0" encoding="iso-8859-1"?>
<decp>
   <cont>
      <vers maj="1.0" min="1.0" rev="" />
      <repris retcr="MAIL" niveau="REPRX" site="C" serve="10.2.128.228"
idform="R_VbIKEJWsR3ZjqABPTMoz8" />
   </cont>
   <dest siren="783973233" siret="78397323300019" ur="U622">
      <lib>URSSAF DE CALAIS</lib>
      <adr1>95 RUE DE VIC</adr1>
      <adr2 />
      <cp>62907</cp>
      <ville>CALAIS CEDEX</ville>
   </dest>
   <coti siret="66655544491000" numc="622000000000091000" cat="4" sscat="31"
etat="A" tldpok="O" dtetat="20070913">
      <nom>EPM1</nom>
      <preno>Jean</preno>
      <adr1>App 101</adr1>
      <adr2>11 rue papin</adr2>
      <cp>62000</cp>
      <ville>Arras</ville>
      <mail />
      <cpladr />
   </coti>
   <moul trsr="N" ech="05" codp="10">
      <dns exo="N" tauxcaf="0.00" plafcaf="0.00">
         <sal code="400" datemb="" datrad="">
            <saladm nir="167095935067449">
               <nnaiss>COUVREUR</nnaiss>
               <nmarit />
               <preno1>Christophe</preno1>
               <preno2 />
               <adr1>2 rue de l yser</adr1>
               <adr2 />
               <cp>59810</cp>
               <ville>LESQUIN</ville>
               <datnai>19670928</datnai>
               <depnai />
               <paynai>000</paynai>
               <vilnai />
            </saladm>
         </sal>
...
```

fichier XSD correspondant

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

<xs:element name="decp">
```

```
 <xs:complexType>
  <xs:all>
   <xs:element ref="cont" minOccurs="1" maxOccurs="1" />
   <xs:element ref="dest" minOccurs="1" maxOccurs="1"/>
   <xs:element ref="coti" minOccurs="1" maxOccurs="1"/>
   <xs:element ref="cotio" minOccurs="0" maxOccurs="1"/>
   <xs:element ref="exi"  minOccurs="1" maxOccurs="1" />
   <xs:element ref="moul" minOccurs="1" maxOccurs="1" />
   <xs:element ref="form" minOccurs="1" maxOccurs="1" />
   <xs:element ref="formo" minOccurs="0" maxOccurs="1" />
  </xs:all>
 </xs:complexType>
</xs:element>
<xs:element name="moul">
  <xs:complexType>
    <xs:all>
      <xs:element name="appel" minOccurs="0" maxOccurs="1">
        <xs:complexType>
          <xs:attribute name="per" use="required" type="xs:integer" />
          <xs:attribute name="mtapp" use="required" type="xs:integer" />
        </xs:complexType>
      </xs:element>
      <xs:element name="tr" minOccurs="0" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="recbrc" minOccurs="0" maxOccurs="1" />
            <xs:element ref="codet" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="efglob" use="required" type="xs:integer" />
          <xs:attribute name="topto" use="required" type="fmtTopTo" />
          <xs:attribute name="mtv2" type="xs:integer" />
        </xs:complexType>
      </xs:element>
      <xs:element name="dec" minOccurs="0" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="codet" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="libcp" type="xs:string" />
                  <xs:element ref="txvar" minOccurs="0" maxOccurs="unbounded" />
                </xs:sequence>
                <xs:attribute name="code" use="required" type="xs:string"/>
                <xs:attribute name="dtef" use="required" type="dtcdType" />
                <xs:attribute name="fmtcp" type="xs:string" />
                <xs:attribute name="cplt" use="required" type="xs:string" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
```

XPATH

Est un langage de requête sur les fichiers XML.

./author	Tous les éléments <author> dans le contexte actuel.
author	Tous les éléments <author> dans le contexte actuel.
first.name	Tous les éléments <first.name> dans le contexte actuel
/bookstore	L'élément de document (<bookstore>) de ce document.
//author	Tous les éléments <author> contenus dans le document.
book[/bookstore/@specialty = @style]	Tous les éléments <book> dont la valeur de l'attribut style est égale à celle de l'attribut specialty de l'élément <bookstore> à la racine du document.
author/first-name	Tous les éléments <first-name> qui sont des enfants d'un élément <author>.
bookstore//title	Tous les éléments <title> à un ou deux niveaux de profondeur dans l'élément <bookstore> (descendants arbitraires). Notez que cette expression est différente de l'expression de la ligne suivante.
bookstore/*/title	Tous les éléments <title> qui sont des petits-enfants d'éléments <bookstore>.
bookstore//book/excerpt//emph	Tous les éléments <emph> situés n'importe où à l'intérieur d'enfants <excerpt> d'éléments <book>, n'importe où à l'intérieur de l'élément <bookstore>.
.//title	Tous les éléments <title> à un ou deux niveaux de profondeur dans le contexte actuel. Notez que cette situation est essentiellement la seule où la notation à points est requise.
author/*	Tous les éléments qui sont les enfants d'éléments <author>.
book/*/last-name	Tous les éléments <last-name> qui sont des petits-enfants d'éléments <book>.
/	Tous les éléments petits-enfants du contexte actuel.
*[@specialty]	Tous les éléments avec l'attribut specialty.
@style	L'attribut style du contexte actuel.
price/@exchange	L'attribut exchange sur des éléments <price> dans le contexte actuel.
price/@exchange/total	Retourne une collection de nœuds vide, car les attributs ne contiennent pas d'éléments enfants. Cette expression est autorisée par la grammaire du langage XML Path (XPath), mais n'est pas valable au sens strict.
book[@style]	Tous les éléments <book> avec des attributs style, dans le contexte actuel.
book/@style	L'attribut style pour tous les éléments <book> du contexte actuel.

@*	Tous les attributs du contexte d'élément actuel.
./first-name	Tous les éléments <first-name> dans le nœud de contexte actuel. Notez que cette expression est équivalente à l'expression de la ligne suivante.
first-name	Tous les éléments <first-name> dans le nœud de contexte actuel.
author[1]	Le premier élément <author> dans le nœud de contexte actuel.
author[first-name][3]	Le troisième élément <author> ayant un enfant <first-name>.
my:book	L'élément <book> de l'espace de noms my.
my:*	Tous les éléments de l'espace de noms my.
@my:*	Tous les attributs de l'espace de noms my (cela ne comprend pas les attributs non qualifiés d'éléments de l'espace de noms my).

```php
<?php
$xml = new domDocument ;
$xml->load('inventory.xml') ;

$xpath = new Domxpath($xml);
$result = $xpath->query("/bookstore/*");
echo $result->length."\n";
foreach ($result as $item) {
      print "#".$item->nodeName."#";
      print "@".$item->getAttribute('style')."#\n";
      if($item->nodeName == "magazine"){
            $item->setAttribute('new','test');
      }
}

echo $xml->saveXML();

?>
```

```xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="myfile.xsl" ?>
<bookstore specialty="novel">
  <book style="autobiography">
    <author>
      <first-name>Joe</first-name>
      <last-name>Bob</last-name>
      <award>Trenton Literary Review Honorable Mention</award>
    </author>
    <price>12</price>
  </book>
  <book style="textbook">
```

```
      <author>
        <first-name>Mary</first-name>
        <last-name>Bob</last-name>
        <publication>Selected Short Stories of
        <first-name>Mary</first-name>
        <last-name>Bob</last-name></publication>
      </author>
      <editor>
        <first-name>Britney</first-name>
        <last-name>Bob</last-name>
      </editor>
      <price>55</price>
    </book>
    <magazine style="glossy" frequency="monthly" new="test">
      <price>2.50</price>
      <subscription price="24" per="year"/>
...
```

XLST

eXtensible Stylesheet Language Transformations, langage de transformation xml.

```php
<?php
$xh = xsl_create();
$result = xslt_process($xh,"inventory.xml","inventory.xsl");
if(!$result){
	echo xslt_error($xh);
}else{
	echo $result."\n" ;
}
xslt_free($xh);
?>
```

```
	histo=0;
	ret_url="/SommaireDecBS.htm";
	periode="<xsl:value-of select="/decp/exi/@per" />";
	cotorga="<xsl:value-of select="/decp/dest/@ur" />";
	numcot="<xsl:value-of select="/decp/coti/@numc" />";
	reel=1;
	test=1;
	<!-- si un RIB est definit -->
	<xsl:if test="/decp/form/pai/tlr/rib/@compt">
	brc=true;
	tep=true;
	iRib=0;
	<xsl:for-each select="/decp/form/pai/tlr/rib">
	tPAI[iRib++] = new PAILine("<xsl:value-of select="@compt" /><xsl:value-of select="@cle"
/>","<xsl:value-of select="@banq" />","<xsl:value-of select="@guich" />",<xsl:value-of
select="@mtrib" />,"<xsl:value-of select="lib" />","EUR");
	</xsl:for-each>
	</xsl:if>
	embeddedbuf="<xsl:call-template name="Replace">
		<xsl:with-param name="chaine" select="$embeddedbuf" />
		<xsl:with-param name="chaineCherche" select="'&#010;'" />
		<xsl:with-param name="chaineRempl" select="'&#092;n'" />
	</xsl:call-template>"

	memp=new employeur ("<xsl:value-of select="decp/coti/@numc" />",
		"<xsl:value-of select="decp/coti/@siret" />",
		2,
		"<xsl:value-of select="decp/coti/nom" />",
		"<xsl:value-of select="decp/coti/preno" />",
		"<xsl:value-of select="decp/coti/adr1" />",
		"<xsl:value-of select="decp/coti/adr2" />",
		"<xsl:value-of select="decp/coti/ville" />",
		"<xsl:value-of select="decp/coti/cp" />",
		"", <!-- date radiation -->
		0, <!-- renseigner -->
		1, <!-- Devise -->
		0, <!-- 0 -->
		<xsl:choose> <!-- 1 -> si code orga U57 ou U67 ou U68, -->
			<xsl:when test="substring(decp/dest/@ur,1,3)='U57'">1</xsl:when>
			<xsl:when test="substring(decp/dest/@ur,1,3)='U67'">1</xsl:when>
			<xsl:when test="substring(decp/dest/@ur,1,3)='U68'">1</xsl:when>
			<xsl:otherwise>0</xsl:otherwise>
		</xsl:choose>,
		"",<!--data->Orga,-->
		"<xsl:value-of select="/decp/coti/mail" />"
	);
...
```