

Accordo su chiavi con OpenSSL

Alfredo De Santis

Dipartimento di Informatica
Università di Salerno

ads@unisa.it

<http://www.dia.unisa.it/professori/ads>



Aprile 2017

DH in OpenSSL

(Generazione dei Parametri)

Mediante il comando **dhparam** è possibile generare i parametri pubblici dello schema DH

Opzioni principali del comando dhparam

```
openssl dhparam [options] [numbits]
```

➤ options

- **-inform arg** Formato di input, dove **arg** può essere DER o PEM
- **-outform arg** Formato di output, dove **arg** può essere DER o PEM
- **-in arg** Dove **arg** è il file di input
- **-out arg** Dove **arg** è il file di output
- **-text** Stampa i parametri Diffie-Hellman in formato testuale
- **-2** Genera i parametri usando 2 come valore del generatore
- **-5** Genera i parametri usando 5 come valore del generatore

➤ numbits

- Numero di bit da generare, di default sono 512

DH in OpenSSL

(Generazione dei Parametri)

Mediante il comando **dhparam** è possibile generare i parametri pubblici dello schema DH

Opzioni principali del comando dhparam

```
openssl dhparam [options] [numbits]
```

➤ options

- **-inform arg** Formato di input per la generazione dei parametri
- **-outform arg** Formato di output per la generazione dei parametri
- **-in arg** Dove **arg** è il file di input
- **-out arg** Dove **arg** è il file di output
- **-text** Stampa i parametri Diffie-Hellman in formato testuale
- **-2** Genera i parametri usando 2 come valore del generatore
- **-5** Genera i parametri usando 5 come valore del generatore

OpenSSL supporta solo questi due valori

➤ Il valore di default è 2

➤ numbits

- Numero di bit da generare, di default sono 512

DH in OpenSSL

(Generazione dei Parametri)

Mediante il comando **dhparam** è possibile generare i parametri pubblici dello schema DH

Opzioni principali del comando dhparam

```
openssl dhparam [options] [numbits]
```

➤ options

- **-inform arg** Formato **arg** out, dove **arg** può essere DER o PEM
- **-outform arg** Formato **arg** out, dove **arg** può essere DER o PEM
- **-in arg** Dove **arg** Per ottenere la lista completa
- **-out arg** Dove **arg** delle opzioni del comando
- **-text** Stampa i parametri **dhparam** è possibile utilizzare **-text** o testuale
- **-2** Genera i parametri **dhparam** **man dhparam** **dhparam** generatore
- **-5** Genera i parametri **dhparam** **dhparam** generatore

➤ numbits

- Numero di bit da generare, di default sono 512

DH in OpenSSL

(File dhparams.pem)

Utilizzando il seguente comando è possibile visualizzare i parametri generati

```
openssl dhparam -in dhparams.pem -text
```

```
Diffie-Hellman-Parameters: (1024 bit)
```

```
prime:
```

```
00:9e:e7:b3:91:2a:2c:6e:ca:38:cd:80:2d:c3:47:
24:14:ed:64:a1:98:45:07:16:4d:e9:e2:2d:1e:84:
15:80:d9:3c:fc:d6:63:db:ff:8f:33:31:36:ef:0d:
e3:9d:7a:af:1c:a2:6a:ca:e7:9e:53:6f:a8:c6:a1:
26:95:ce:17:ea:9d:cd:c8:11:69:21:e2:2c:bd:25:
fe:20:dc:9f:49:c5:33:58:fd:8d:7c:07:57:6c:1c:
5a:b3:73:45:22:4d:ef:d5:34:b3:ac:a6:5d:a3:04:
81:13:2a:a4:a7:4e:34:60:1b:73:b6:0a:3b:a0:3d:
4d:d7:81:4a:f3:39:6d:67:a3
```

```
generator: 2 (0x2)
```

```
-----BEGIN DH PARAMETERS-----
```

```
MIGHAoGBAJ7ns5EqLG7KOM2ALcNHJBTtZKGYRQcWTeniLR6EFYDZPPzWY9v/jzMx
Nu8N4516rxyiasrnnlNvqMahJpX0F+qdzcgRaSHiLL0l/iDcn0nFM1j9jXwHV2wc
WrNzRSJN79U0s6ymXaMEgRMqKdONGAbc7YK06A9TdeBSvM5bWejAgEC
```

```
-----END DH PARAMETERS-----
```

Codifica PEM
dei parametri
DH

p

g

DH in OpenSSL

(File dhparams.pem)

Utilizzando il seguente comando è

```
openssl dhparam
```

I parametri e le chiavi DH sono rappresentati e codificati secondo gli standard PKCS #3 ed ANSI X9.42

ti

```
Diffie-Hellman-Parameter (1024 bit)
```

```
prime:
```

```
00:9e:e7:b3:91:2a:2c:6e:ca:38:cd:80:2d:c3:47:
24:14:ed:64:a1:98:45:07:16:4d:e9:e2:2d:1e:84:
15:80:d9:3c:fc:d6:63:db:ff:8f:33:31:36:ef:0d:
e3:9d:7a:af:1c:a2:6a:ca:e7:9e:53:6f:a8:c6:a1:
26:95:ce:17:ea:9d:cd:c8:11:69:21:e2:2c:bd:25:
fe:20:dc:9f:49:c5:33:58:fd:8d:7c:07:57:6c:1c:
5a:b3:73:45:22:4d:ef:d5:34:b3:ac:a6:5d:a3:04:
81:13:2a:a4:a7:4e:34:60:1b:73:b6:0a:3b:a0:3d:
4d:d7:81:4a:f3:39:6d:67:a3
```

p

g

```
generator: 2 (0x2)
```

```
-----BEGIN DH PARAMETERS-----
```

```
MIGHAoGBAJ7ns5EqLG7KOM2ALcNHJBTtZKGYRQcWTeniLR6EFYDZPPzWY9v/jzMx
Nu8N4516rxyiasrnnlNvqMahJpXOF+qdzcgRaSHiLL0l/iDcn0nFM1j9jXwHV2wc
WrNzRSJN79U0s6ymXaMEgRMqKdONGAbc7YK06A9TdeBSvM5bWejAgEC
```

```
-----END DH PARAMETERS-----
```

Codifica PEM
dei parametri
DH

DH in OpenSSL

(Esempio Generazione Chiavi)

Ogni utente utilizza i parametri pubblici per generare la propria coppia di chiavi (privata e pubblica), memorizzandola in un file

➤ Assumiamo che tale file sia `dhkey1.pem` (per l'**Utente 1**) e `dhkey2.pem` (per l'**Utente 2**)

Utente 1

```
openssl genpkey -paramfile dhparams.pem -out dhkey1.pem
```

Utente 2

```
openssl genpkey -paramfile dhparams.pem -out dhkey2.pem
```

È possibile visualizzare la struttura di un file contenente una coppia di chiavi (ad es., `dhkey1.pem`) mediante il seguente comando

```
openssl pkey -in dhkey1.pem -text
```


DH in OpenSSL

(Esempio Generazione Chiavi)

Ogni utente utilizza
chiavi (privata e pubblica)
➤ Assumiamo che
(per l'Utente 2)

Per ottenere la lista completa
delle opzioni del comando
`genpkey` è possibile utilizzare
`man genpkey`

generare la propria coppia di
in un file
per l'Utente 1) e `dhkey2.pem`

Utente 1

```
openssl genpkey -paramfile dhparams.pem -out dhkey1.pem
```

Utente 2

```
openssl genpkey -paramfile dhparams.pem -out dhkey2.pem
```

Per ottenere la lista completa
delle opzioni del comando `pkey`
è possibile utilizzare
`man pkey`

È possibile visualizzare la struttura di una coppia di
chiavi (ad es., `dhkey1.pem`) mediante il seguente comando

```
openssl pkey -in dhkey1.pem -text
```

DH in OpenSSL

(Contenuto del file dhkey1.pem)

DH Private-Key: (1024 bit)

private-key:

```
40:51:79:87:a0:af:d4:28:48:e0:b4:64:61:a8:09:
8d:26:8d:a7:0c:37:66:cc:ce:37:07:98:79:6f:46:
ec:39:e1:1c:fc:3d:af:29:48:fc:7a:3b:34:ab:c2:
e8:f3:fb:43:17:a8:7c:c7:16:ca:07:9a:54:27:06:
34:06:78:03:32:c2:b0:71:33:de:af:17:1a:85:10:
43:60:38:71:76:b2:f2:94:cc:8f:b7:4c:b9:9d:8a:
66:9a:85:4c:c6:8f:8a:a7:11:bb:79:40:d0:ff:0d:
02:dc:97:6c:92:09:6e:5a:9e:c8:3d:80:43:f3:a5:
f6:03:12:47:16:4a:a4:60
```

Chiave Privata (x)

public-key:

```
00:85:14:d3:dc:3b:ed:83:3d:46:09:c9:a7:14:f5:
94:d2:c1:87:b5:10:0d:39:79:7a:c5:a6:9d:be:e0:
94:c5:6d:4a:4e:ee:f9:09:69:32:dc:1b:79:4d:a0:
d9:a3:5b:9e:c2:e8:bf:e6:4f:2d:a9:ea:79:ed:85:
ce:64:d2:1f:a8:15:4b:f6:73:72:6f:fc:cc:bf:45:
ea:0a:71:4d:e8:28:9f:cd:1d:21:90:a5:b3:54:d8:
3b:87:68:09:89:aa:3d:5a:b0:ce:8b:40:ad:2d:23:
b2:20:a2:6f:2a:a9:ae:9d:a2:80:1b:0d:bd:f0:7d:
9c:80:62:30:7b:69:3c:8a:53
```

Chiave Pubblica ($g^x \text{ mod } p$)

prime:

```
00:8c:39:39:d1:60:1a:c7:cb:cd:7a:1a:f7:8f:db:
22:cc:97:76:d1:a4:c9:c7:d8:f4:98:a3:3f:4d:24:
17:9f:97:a9:e1:9a:ed:e6:3c:3d:36:16:6e:2a:a8:
00:04:09:9c:9d:13:32:c2:01:8e:7a:e5:eb:49:6c:
0f:52:47:5b:51:d2:89:e3:9b:0d:f4:cf:fb:a4:a8:
3f:64:ec:73:74:63:4b:b9:c9:2a:ec:ee:e1:cf:3c:
67:3d:21:c4:f1:96:b9:09:85:39:34:bb:14:ce:a2:
38:c7:33:59:37:11:c7:fe:0d:99:e0:7a:59:4e:7d:
dc:bd:28:2a:01:16:f7:87:13
```

Numero Primo (p)

generator: 2 (0x2)

Generatore (g)

DH in OpenSSL

(Esempio Esportazione Chiavi Pubbliche)

Gli utenti devono scambiarsi le loro rispettive chiavi pubbliche

- Ciascun utente deve estrarre la propria chiave pubblica e memorizzarla in un apposito file

Utente 1

```
openssl pkey -in dhkey1.pem -pubout -out dhpublish1.pem
```

Utente 2

```
openssl pkey -in dhkey2.pem -pubout -out dhpublish2.pem
```

Per visualizzare la struttura del file `dhpublish1.pem`

```
openssl pkey -pubin -in dhpublish1.pem -text
```

DH in OpenSSL

(Contenuto del file dhpub1.pem)

```
-----BEGIN PUBLIC KEY-----  
MIIBIDCB1QYJKoZIhvcNAQMBMIGHAoGBAIw50dFgGs fLzXoa94/bIsyXdtGkycfY  
9JijP00kF5+XqeGa7eY8PTYWbiquAAQJnJ0TMsIBjnrl60lsD1JHW1HSie0bDfTP  
+6SoP2Tsc3Rjs7nJKuzu4c88Zz0hxPGWuQmF0TS7FM6i0MczWTcRx/4NmeB6WU59  
3L0oKgEW94cTAgECA4GFAAKBgQCFNPc0+2DPUYJyacU9ZTSwYe1EA05eXrFpp2+  
4JTfBUp07vkJaTLCg3lNoNmjW57C6L/mTy2p6nnthc5k0h+oFUv2c3Jv/My/ReoK  
cU3oKJ/NHSGQpbNU2DuHaAmJqj1asM6LQK0tI7Igom8qqa6dooAbDb3wfZyAYjB7  
aTyKUw==
```

```
-----END PUBLIC KEY-----  
DH Public-Key: (1024 bit)
```

```
public-key:
```

```
00:85:14:d3:dc:3b:ed:83:3d:46:09:c9:a7:14:f5:  
94:d2:c1:87:b5:10:0d:39:79:7a:c5:a6:9d:be:e0:  
94:c5:6d:4a:4e:ee:f9:09:69:32:dc:1b:79:4d:a0:  
d9:a3:5b:9e:c2:e8:bf:e6:4f:2d:a9:ea:79:ed:85:  
ce:64:d2:1f:a8:15:4b:f6:73:72:6f:fc:cc:bf:45:  
ea:0a:71:4d:e8:28:9f:cd:1d:21:90:a5:b3:54:d8:  
3b:87:68:09:89:aa:3d:5a:b0:ce:8b:40:ad:2d:23:  
b2:20:a2:6f:2a:a9:ae:9d:a2:80:1b:0d:bd:f0:7d:  
9c:80:62:30:7b:69:3c:8a:53
```

Chiave Pubblica ($g^x \text{ mod } p$)

```
prime:
```

```
00:8c:39:39:d1:60:1a:c7:cb:cd:7a:1a:f7:8f:db:  
22:cc:97:76:d1:a4:c9:c7:d8:f4:98:a3:3f:4d:24:  
17:9f:97:a9:e1:9a:ed:e6:3c:3d:36:16:6e:2a:a8:  
00:04:09:9c:9d:13:32:c2:01:8e:7a:e5:eb:49:6c:  
0f:52:47:5b:51:d2:89:e3:9b:0d:f4:cf:fb:a4:a8:  
3f:64:ec:73:74:63:4b:b9:c9:2a:ec:ee:e1:cf:3c:  
67:3d:21:c4:f1:96:b9:09:85:39:34:bb:14:ce:a2:  
38:c7:33:59:37:11:c7:fe:0d:99:e0:7a:59:4e:7d:  
dc:bd:28:2a:01:16:f7:87:13
```

Numero Primo (p)

```
generator: 2 (0x2)
```

Generatore (g)

DH in OpenSSL

(Esempio Calcolo del Segreto Condiviso)

Sia l'**Utente 1** che l'**Utente 2** eseguono i seguenti comandi per ottenere la *chiave condivisa*, composta da 1014 bit, che verrà memorizzato nei file **segreto1.bin** e **segreto2.bin**, rispettivamente

Utente 1

```
openssl pkeyutl -derive -inkey dhkey1.pem -peerkey dhp2b2.pem -out  
segreto1.bin
```

Utente 2

```
openssl pkeyutl -derive -inkey dhkey2.pem -peerkey dhp1b1.pem -out  
segreto2.bin
```

Se tutto andrà a buon fine, **segreto1.bin** e **segreto2.bin** saranno esattamente identici

➤Ciò può essere verificato mediante gli appositi comandi forniti da ciascun Sistema Operativo, ad es.

```
cmp -b segreto1.bin segreto2.bin
```

DH in OpenSSL

(Esempio Calcolo del Segreto Condiviso)

Sia l'**Utente 1** che l'**Utente 2** vogliono ottenere la lista completa delle opzioni del comando `openssl pkeyutl` è possibile utilizzare il comando `man pkeyutl`.
Per ottenere la lista completa delle opzioni del comando `openssl pkeyutl` è possibile utilizzare il comando `man pkeyutl`.
I seguenti comandi per ottenere la lista completa delle opzioni del comando `openssl pkeyutl` che verrà memorizzato nei file `segreto1.bin` e `segreto2.bin` rispettivamente.

Utente 1

```
openssl pkeyutl -derive -inkey dhkey1.pem -peerkey dhp2b2.pem -out segreto1.bin
```

Utente 2

```
openssl pkeyutl -derive -inkey dhkey2.pem -peerkey dhp1b1.pem -out segreto2.bin
```

Se tutto andrà a buon fine, **segreto1.bin** e **segreto2.bin** saranno esattamente identici

➤ Ciò può essere verificato mediante gli appositi comandi forniti da ciascun Sistema Operativo, ad es.

```
cmp -b segreto1.bin segreto2.bin
```

DH in OpenSSL

(Esempio Calcolo del Segreto Condiviso)

Sia l'**Utente 1** che l'**Utente 2** hanno generato una *chiave condivisa*, comparsa in **segreto1.bin** e **segreto2.bin**

Utente 1

```
openssl pkeyutl -de  
segreto1.bin
```

Utente 2

```
openssl pkeyutl -de  
segreto2.bin
```

```
NAME  
cmp - compare two files byte by byte  
  
SYNOPSIS  
cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]  
  
DESCRIPTION  
Compare two files byte by byte.  
  
-b --print-bytes  
    Print differing bytes.  
  
-i SKIP --ignore-initial=SKIP  
    Skip the first SKIP bytes of input.  
  
-i SKIP1:SKIP2 --ignore-initial=SKIP1:SKIP2  
    Skip the first SKIP1 bytes of FILE1 and the first SKIP2 bytes of  
    FILE2.  
  
-l --verbose  
    Output byte numbers and values of all differing bytes.  
  
man cmp
```

Se tutto andrà a buon fine, i file **segreto1.bin** e **segreto2.bin** saranno esattamente identici

➤ Ciò può essere verificato mediante gli appositi comandi forniti da ciascun Sistema Operativo, ad es.

```
cmp -b segreto1.bin segreto2.bin
```

DH in OpenSSL

(Esempio Calcolo del Segreto Condiviso)

Sia l'**Utente 1** che l'**Utente 2** eseguono i seguenti comandi per ottenere la *chiave condivisa*, composta da 1014 bit, che verrà memorizzato nei file **segreto1.bin** e **segreto2.bin**, rispettivamente

Utente 1

```
openssl pkeyutl -derive -inkey dhkey1.pem -peerkey dhp2b2.pem -out  
segreto1.bin
```

Utente 2

```
openssl pkeyutl -derive -inkey dhkey2.pem -peerkey dhp1b1.pem -out  
segreto2.bin
```

Se tutto andrà a buon fine, i file saranno esattamente identici

➤ Ciò può essere verificato mediante gli appositi comandi forniti da ciascun Sistema Operativo, ad esempio:

Tale comando non restituisce nulla in output se i file comparati sono identici

```
cmp -b segreto1.bin segreto2.bin
```


DH in OpenSSL

(File `segreto1.bin` e `segreto2.bin`)

```
$ xxd segreto1.bin
00000000: 4e42 91a9 a3e2 5af8 9749 6f1d 38aa 5345  NB....Z..Io.8.SE
00000010: 4a9d 6ab7 82e6 03b1 d6f8 cc53 690d 2a98  J.j.....Si.*.
00000020: 2599 33e2 8eac 4b5b 5ace a565 91b7 27a8  %.3...K[Z..e..'
00000030: 280c dae8 cd3e 0b9e b876 8d7a 9b65 d860  (...>...v.z.e.`
00000040: f7b2 d14b df63 23a9 b1bf 0016 b9d8 8c22  ...K.c#....."
00000050: 4640 7c4f 2c4f 5ad3 dc4e 498d eac5 7d6b  F@|0,0Z..NI...}k
00000060: f896 256b 12a1 f29c 1514 8964 6689 897d  ..%k.....df..}
00000070: 2b1e a34a 968f 2f6e 0c66 ff78 df1b 67b1  +..J../n.f.x..g.
```

```
$ xxd segreto2.bin
00000000: 4e42 91a9 a3e2 5af8 9749 6f1d 38aa 5345  NB....Z..Io.8.SE
00000010: 4a9d 6ab7 82e6 03b1 d6f8 cc53 690d 2a98  J.j.....Si.*.
00000020: 2599 33e2 8eac 4b5b 5ace a565 91b7 27a8  %.3...K[Z..e..'
00000030: 280c dae8 cd3e 0b9e b876 8d7a 9b65 d860  (...>...v.z.e.`
00000040: f7b2 d14b df63 23a9 b1bf 0016 b9d8 8c22  ...K.c#....."
00000050: 4640 7c4f 2c4f 5ad3 dc4e 498d eac5 7d6b  F@|0,0Z..NI...}k
00000060: f896 256b 12a1 f29c 1514 8964 6689 897d  ..%k.....df..}
00000070: 2b1e a34a 968f 2f6e 0c66 ff78 df1b 67b1  +..J../n.f.x..g.
```

DH in OpenSSL

(File `segreto1.bin` e `segreto2.bin`)

```
$ xxd segreto1.bin
00000000: 4e42 91a9 a3e2 5af8 9749 6f1d 38aa 5345  NB....Z..Io.8.SE
00000010: 1a9d 6ab7 82e6 03b1 d6f8 cc53 690d 2a98  J.j.....Si.*.
00000020: 33e2 8eac 4b5b 5ace a565 91b7 27a8  %.3...K[Z..e..'
00000030: 8 cd3e 0b9e b876 8d7a 9b65 d860  (...>...v.z.e.`
00000040: f 563 23a9 b1bf 0016 b9d8 8c22  ...K.c#....."
00000050: 464 5ad3 dc4e 498d eac5 7d6b  F@|0,0Z..NI...}k
00000060: f896 1514 8964 6689 897d  ..%k.....df..}
```

00 NAME

`xxd` - make a hexdump or do the reverse.

\$ SYNOPSIS

```
00 xxd -h[elp]
00 xxd [options] [infile [outfile]]
00 xxd -r[everse] [options] [infile [outfile]]
```

00 DESCRIPTION

00 `xxd` creates a hex dump of a given file or standard input. It can also convert a
00 hex dump back to its original binary form. Like `uuencode(1)` and `uudecode(1)` it
00 allows the transmission of binary data in a 'mail-safe' ASCII representation, but
00 has the advantage of decoding to standard output. Moreover, it can be used to
00 perform binary file patching.

DH in OpenSSL

(File segreto1.bin)

00000000	4E 42 91 A9 A3 E2 5A F8 97 49 6F 1D 38 AA 53 45 4A 9D	NB...Z..Io.8.SEJ.
00000012	6A B7 82 E6 03 B1 D6 F8 CC 53 69 0D 2A 98 25 99 33 E2	j.....Si.*.%3.
00000024	8E AC 4B 5B 5A CE A5 65 91 B7 27 A8 28 0C DA E8 CD 3E	..K[Z..e..'.(....>
00000036	0B 9E B8 76 8D 7A 9B 65 D8 60 F7 B2 D1 4B DF 63 23 A9	...v.z.e.`...K.c#.
00000048	B1 BF 00 16 B9 D8 8C 22 46 40 7C 4F 2C 4F 5A D3 DC 4E "F@ O,OZ..N
0000005a	49 8D EA C5 7D 6B F8 96 25 6B 12 A1 F2 9C 15 14 89 64	I...}k..%k.....d
0000006c	66 89 89 7D 2B 1E A3 4A 96 8F 2F 6E 0C 66 FF 78 DF 1B	f..}+..J../n.f.x..
0000007e	67 B1	g.

Signed 8 bit: Signed 32 bit: Hexadecimal: ✕

Unsigned 8 bit: Unsigned 32 bit: Decimal:

Signed 16 bit: Float 32 bit: Octal:

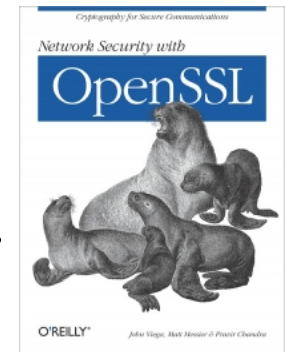
Unsigned 16 bit: Float 64 bit: Binary:

Show little endian decoding Show unsigned as hexadecimal ASCII Text:

Offset: 0 / 127 Selection: 0 to 127 (128 bytes) INS

Bibliografia

- **Network Security with OpenSSL**
Pravir Chandra, Matt Messier and John Viega (2002), O'Reilly
 - Cap. 2.4.1
 - Appendix A. Command-Line Reference



- **Documentazione su OpenSSL**
 - <https://www.openssl.org/docs/>

Domande?

