Diffie Hellman Secret Key Exchange using OpenSSL

By Steven Gordon on Sun, 27/01/2013 - 10:24am

An example of using OpenSSL operations to perform a Diffie-Hellmen secret key exchange (DHKE). The goal in DHKE is for two users to obtain a shared secret key, without any other users knowing that key. The exchange is performed over a public network, i.e. all messages sent between the two users can be intercepted and read by any other user. The protocol makes use of modular arithmetic and especially exponentials. The security of the protocol relies on the fact that solving a discrete logarithm (the inverse of an exponential) is practically impossible when large enough values are used.

Wikipedia has a description and example of DHKE [3]. My lecture slides [4] on public key cryptography also include a description. My description of DHKE starts at about 39m 30s into the screencast available on YouTube [5]. It includes a simple example starting at 47m 53s. View below to go straight to the DHKE portion of the lecture.

OpenSSL [6] provides both a library of security operations you can access from your own software, as well as a command line mode. In the past I've given examples of using OpenSSL to generate RSA keys [7] as well as encrypt and sign with RSA [8]. In the following I demonstrate using OpenSSL for DHKE.

DHKE is performed by two users, on two different computers. For my demo I do everything on one computer. The steps performed by each user are the same, but just with different files. In the following there is user 1 and user 2.

0.1 Steps for Diffie-Hellman Key Exchange with OpenSSL

Generate the Diffie-Hellman global public parameters, saving them in the file dhp.pem [9]:

```
$ openssl genpkey -genparam -algorithm DH -out dhp.pem
...+.....................................................................................
```

Display the generated global public parameters, first in the encoded form, then in the text form:

```
$ cat dhp.pem
-----BEGIN DH PARAMETERS-----
MIGHAoGBAOZVzJ4E8766527Mp3FD71xEUYdmFan4tPcSuPO99H7n9xfAm7WytmRQ
gxNn2dz4X58FKLzVMY+x2rLyPOd8SLa3OB7tE+gKFMymswteN//lPbFeLWtyei78
7lGJNnjVDpqJFmo1nldMTDyl5Z+ueZJP5vGGs2ouvem/Cf5N5QRTAgEC
-----END DH PARAMETERS-----
```

```
$ openssl pkeyparam -in dhp.pem -text
-----BEGIN DH PARAMETERS-----
MIGHAoGBAOZVzJ4E8766527Mp3FD71xEUYdmFan4tPcSuPO99H7n9xfAm7WytmRQ
gxNn2dz4X58FKLzVMY+x2rLyPOd8SLa3OB7tE+gKFMymswteN//lPbFeLWtyei78
7lGJNnjVDpqJFmo1nldMTDyl5Z+ueZJP5vGGs2ouvem/Cf5N5QRTAgEC
-----END DH PARAMETERS-----
PKCS#3 DH Parameters: (1024 bit)
    prime:
        00:e6:55:cc:9e:04:f3:be:ba:e7:6e:cc:a7:71:43:
        ef:5c:44:51:87:66:15:a9:f8:b4:f7:12:b8:f3:bd:
        f4:7e:e7:f7:17:c0:9b:b5:b2:b6:64:50:83:13:67:
        d9:dc:f8:5f:9f:05:28:bc:d5:31:8f:b1:da:b2:f2:
        3c:e7:7c:48:b6:b7:38:1e:ed:13:e8:0a:14:cc:a6:
        b3:0b:5e:37:ff:e5:3d:b1:5e:2d:6b:72:7a:2e:fc:
        ee:51:89:36:78:d5:0e:9a:89:16:6a:35:9e:57:4c:
        4c:3c:a5:e5:9f:ae:79:92:4f:e6:f1:86:b3:6a:2e:
        bd:e9:bf:09:fe:4d:e5:04:53
    generator: 2 (0x2)
```

Each user now uses the public parameters to generate their own private and public key, saving them in the file dhkey1.pem [10] (for user 1) and dhkey2.pem [11] (for user 2):

```
$ openssl genpkey -paramfile dhp.pem -out dhkey1.pem
```

```
$ openssl pkey -in dhkey1.pem -text -noout
PKCS#3 DH Private-Key: (1024 bit)
    private-key:
        48:88:7d:fd:09:0d:17:5e:33:be:ea:29:e7:b3:83:
        34:29:92:89:06:9f:9a:b4:92:b6:78:07:90:5f:aa:
        98:d9:6d:22:d7:92:05:be:f0:3f:14:af:09:3f:17:
        97:b9:04:73:41:32:c3:4a:38:8f:dc:79:e2:04:97:
        bf:a1:46:5f:ec:2a:ac:4f:ab:df:3b:b0:c9:be:86:
```

```
        85:d2:0f:7b:fe:03:46:a9:ab:df:7f:a8:98:38:c3:
        fa:9c:a6:ab:db:70:be:a6:67:95:ab:66:99:cc:15:
        4d:b5:94:90:e4:15:9f:14:2f:7b:dd:ff:60:3c:1d:
        3d:6c:4f:ff:81:77:e1:1d
    public-key:
        00:d9:ab:d7:8c:93:df:dd:eb:92:0d:57:d6:51:31:
        26:d8:f1:11:8c:92:37:a4:51:01:40:8d:bf:fe:6c:
        fd:95:b0:11:a0:16:e4:e0:ab:8a:ef:06:01:e8:36:
        a4:52:b8:bb:88:be:7c:a7:1e:4f:22:f9:7a:a6:5f:
        83:58:ee:69:34:8d:12:27:d6:5d:b6:e5:36:41:d1:
        a6:54:2a:a4:be:4b:4a:dc:75:fa:c8:16:af:79:a8:
        e3:f5:09:7f:83:13:e7:b7:25:df:37:ea:dc:8c:77:
        4e:20:33:df:a9:9c:95:cc:ef:33:3b:f4:02:b0:66:
        19:8c:30:48:1e:2a:83:87:5c
    prime:
        00:e6:55:cc:9e:04:f3:be:ba:e7:6e:cc:a7:71:43:
        ef:5c:44:51:87:66:15:a9:f8:b4:f7:12:b8:f3:bd:
        f4:7e:e7:f7:17:c0:9b:b5:b2:b6:64:50:83:13:67:
        d9:dc:f8:5f:9f:05:28:bc:d5:31:8f:b1:da:b2:f2:
        3c:e7:7c:48:b6:b7:38:1e:ed:13:e8:0a:14:cc:a6:
        b3:0b:5e:37:ff:e5:3d:b1:5e:2d:6b:72:7a:2e:fc:
        ee:51:89:36:78:d5:0e:9a:89:16:6a:35:9e:57:4c:
        4c:3c:a5:e5:9f:ae:79:92:4f:e6:f1:86:b3:6a:2e:
        bd:e9:bf:09:fe:4d:e5:04:53
    generator: 2 (0x2)
```

The other user uses the same public parameters, `dhp.pem`, to generate their private/public key:

```
$ openssl genpkey -paramfile dhp.pem -out dhkey2.pem
```

```
$ openssl pkey -in dhkey2.pem -text -noout
PKCS#3 DH Private-Key: (1024 bit)
    private-key:
        5d:70:9b:3e:a7:c9:b1:3b:df:17:d3:76:dd:45:f0:
        38:6d:be:35:f6:79:5d:05:bf:e2:63:b0:ea:25:00:
        61:0a:4c:e2:e4:e7:8e:97:6e:cb:9e:f0:f9:4b:d9:
        1c:2e:d6:b1:71:cb:ec:56:a7:2f:b0:af:ff:67:df:
        37:e0:d8:8c:ab:5d:ef:3d:27:c5:5a:a6:8d:49:30:
        6b:4e:d4:1f:5c:40:da:35:d0:bc:c7:3d:16:a3:13:
        2e:86:af:13:8b:65:c4:19:f2:75:43:e7:11:b6:5a:
        81:d1:e0:ff:5d:f3:c2:f4:6f:d2:f0:72:97:66:b9:
        93:3d:17:b0:06:ef:8a:3b
    public-key:
        00:d9:9a:00:1b:98:f5:0b:e2:d6:57:f7:4d:e3:4b:
        aa:43:ad:e2:f2:93:31:a1:e7:4b:a7:06:dc:ab:22:
        09:5a:0d:41:1a:c1:37:c0:6d:88:f4:7c:0a:22:27:
        1e:d3:84:39:51:92:62:d5:14:9e:68:ee:2f:69:27:
        ae:dd:d1:e6:a2:5f:3c:d2:7b:a7:7c:8e:61:28:fb:
        8b:1c:d7:a0:0b:d3:7b:37:af:78:b2:7e:eb:62:a7:
        85:b6:0f:90:10:b7:9c:ce:ec:84:a9:28:e3:7f:22:
        8f:76:cd:68:58:56:45:fd:3e:36:37:a1:99:aa:ca:
        4a:65:65:af:a8:21:ee:1f:b6
    prime:
        00:e6:55:cc:9e:04:f3:be:ba:e7:6e:cc:a7:71:43:
        ef:5c:44:51:87:66:15:a9:f8:b4:f7:12:b8:f3:bd:
        f4:7e:e7:f7:17:c0:9b:b5:b2:b6:64:50:83:13:67:
```

```
        d9:dc:f8:5f:9f:05:28:bc:d5:31:8f:b1:da:b2:f2:
        3c:e7:7c:48:b6:b7:38:1e:ed:13:e8:0a:14:cc:a6:
        b3:0b:5e:37:ff:e5:3d:b1:5e:2d:6b:72:7a:2e:fc:
        ee:51:89:36:78:d5:0e:9a:89:16:6a:35:9e:57:4c:
        4c:3c:a5:e5:9f:ae:79:92:4f:e6:f1:86:b3:6a:2e:
        bd:e9:bf:09:fe:4d:e5:04:53
    generator: 2 (0x2)
```

The users must exchange their public keys. First extract the public key into the file dhpub1.pem [12] (and similar user 2 creates dh2pub.pem [13] - this step is not shown below):

```
$ openssl pkey -in dhkey1.pem -pubout -out dhpub1.pem
```

```
$ openssl pkey -pubin -in dhpub1.pem -text
-----BEGIN PUBLIC KEY-----
MIIBIDCBlQYJKoZIhvcNAQMBMIGHAoGBAOZVzJ4E8766527Mp3FD71xEUYdmFan4
tPcSuPO99H7n9xfAm7WytmRQgxNn2dz4X58FKLzVMY+x2rLyPOd8SLa3OB7tE+gK
FMymswteN//lPbFeLWtyei787lGJNnjVDpqJFmo1nldMTDyl5Z+ueZJP5vGGs2ou
vem/Cf5N5QRTAgECA4GFAAKBgQDZq9eMk9/d65INV9ZRMSbY8RGMkjekUQFAjb/+
bP2VsBGgFuTgq4rvBgHoNqRSuLuIvnynHk8i+XqmX4NY7mk0jRIn1l225TZB0aZU
KqS+S0rcdfrIFq95qOP1CX+DE+e3Jd836tyMd04gM9+pnJXM7zM79AKwZhmMMEge
KoOHXA==
-----END PUBLIC KEY-----
PKCS#3 DH Public-Key: (1024 bit)
    public-key:
        00:d9:ab:d7:8c:93:df:dd:eb:92:0d:57:d6:51:31:
        26:d8:f1:11:8c:92:37:a4:51:01:40:8d:bf:fe:6c:
        fd:95:b0:11:a0:16:e4:e0:ab:8a:ef:06:01:e8:36:
        a4:52:b8:bb:88:be:7c:a7:1e:4f:22:f9:7a:a6:5f:
        83:58:ee:69:34:8d:12:27:d6:5d:b6:e5:36:41:d1:
        a6:54:2a:a4:be:4b:4a:dc:75:fa:c8:16:af:79:a8:
        e3:f5:09:7f:83:13:e7:b7:25:df:37:ea:dc:8c:77:
        4e:20:33:df:a9:9c:95:cc:ef:33:3b:f4:02:b0:66:
        19:8c:30:48:1e:2a:83:87:5c
    prime:
        00:e6:55:cc:9e:04:f3:be:ba:e7:6e:cc:a7:71:43:
        ef:5c:44:51:87:66:15:a9:f8:b4:f7:12:b8:f3:bd:
        f4:7e:e7:f7:17:c0:9b:b5:b2:b6:64:50:83:13:67:
        d9:dc:f8:5f:9f:05:28:bc:d5:31:8f:b1:da:b2:f2:
        3c:e7:7c:48:b6:b7:38:1e:ed:13:e8:0a:14:cc:a6:
        b3:0b:5e:37:ff:e5:3d:b1:5e:2d:6b:72:7a:2e:fc:
        ee:51:89:36:78:d5:0e:9a:89:16:6a:35:9e:57:4c:
        4c:3c:a5:e5:9f:ae:79:92:4f:e6:f1:86:b3:6a:2e:
        bd:e9:bf:09:fe:4d:e5:04:53
    generator: 2 (0x2)
```

After exchanging public keys, i.e. the files `dhpub1.pem` and `dhpub2.pem`, each user can derive the shared secret. User 1 performs the following to output the secret, a 128 Byte binary value into the file secret1.bin [14]:

```
$ openssl pkeyutl -derive -inkey dhkey1.pem -peerkey dhpub2.pem -out secret1.bin
```

The other user does the same using their private key and user 1's public key to produce

secret2.bin [15]:

```
$ openssl pkeyutl -derive -inkey dhkey2.pem -peerkey dhpub1.pem -out secret2.bin
```

The secrets should be the same:

```
$ cmp secret1.bin secret2.bin
$ xxd secret1.bin
0000000: b7cb b892 b541 7810 d8ec d089 6c89 3c19  .....Ax.....l.<.
0000010: e8e1 27d8 66ee dac8 684a f0bd 0a7f e7d3  ..'.f...hJ......
0000020: 3643 8654 fddf 4399 e58e 2c7c 3d33 9532  6C.T..C...,|=3.2
0000030: f693 edf2 c9a0 40e8 58b8 38de 74a5 c0b0  ......@.X.8.t...
0000040: 64ab 4006 a3cd d795 2cef d0fc 2b0f d1ab  d.@.....,...+...
0000050: d1e5 1a2a 3431 e3fa ba63 f7cf 1c61 ff65  ...*41...c...a.e
0000060: d9cd c85d c5fe 5c50 c543 aaeb de49 8501  ...]..\P.C...I..
0000070: 6cf1 66a6 87b6 ddec 835c b4b1 3d9d e2fe  l.f......\..=...
$ xxd secret2.bin
0000000: b7cb b892 b541 7810 d8ec d089 6c89 3c19  .....Ax.....l.<.
0000010: e8e1 27d8 66ee dac8 684a f0bd 0a7f e7d3  ..'.f...hJ......
0000020: 3643 8654 fddf 4399 e58e 2c7c 3d33 9532  6C.T..C...,|=3.2
0000030: f693 edf2 c9a0 40e8 58b8 38de 74a5 c0b0  ......@.X.8.t...
0000040: 64ab 4006 a3cd d795 2cef d0fc 2b0f d1ab  d.@.....,...+...
0000050: d1e5 1a2a 3431 e3fa ba63 f7cf 1c61 ff65  ...*41...c...a.e
0000060: d9cd c85d c5fe 5c50 c543 aaeb de49 8501  ...]..\P.C...I..
0000070: 6cf1 66a6 87b6 ddec 835c b4b1 3d9d e2fe  l.f......\..=...
```

Interest: Ubuntu Linux [16]
OpenSSL [17]
Topic: Security [18]
Content: Howto [19]

Source URL: http://sandilands.info/sgordon/diffie-hellman-secret-key-exchange-with-openssl

Links:
[1] http://sandilands.info/sgordon/diffie-hellman-secret-key-exchange-with-openssl
[2] http://sandilands.info/sgordon/user/2
[3] http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
[4] http://ict.siit.tu.ac.th/~sgordon/css322y12s2/lectures.html#pkc
[5] http://www.youtube.com/watch?v=nWgxdSdfDLY
[6] http://www.openssl.com/
[7] http://sandilands.info/sgordon/key-generation-and-encryption-examples-using-openssl
[8] http://sandilands.info/sgordon/public-key-encryption-and-digital-signatures-using-openssl
[9] http://sandilands.info/sgordon/doc/security/dhp.pem
[10] http://sandilands.info/sgordon/doc/security/dhkey1.pem
[11] http://sandilands.info/sgordon/doc/security/dhkey2.pem
[12] http://sandilands.info/sgordon/doc/security/dhpub1.pem
[13] http://sandilands.info/sgordon/doc/security/dhpub2.pem
[14] http://sandilands.info/sgordon/doc/security/secret1.bin
[15] http://sandilands.info/sgordon/doc/security/secret2.bin
[16] http://sandilands.info/sgordon/taxonomy/term/302
[17] http://sandilands.info/sgordon/taxonomy/term/338
[18] http://sandilands.info/sgordon/taxonomy/term/116
[19] http://sandilands.info/sgordon/taxonomy/term/212