

Cryptography

Agreeing a Secure Session Key in Public (Without Disclosing this Key to any Listening 3rd Party)

Steve Cholerton

When setting up a secure connection across the internet, for example between your browser and your online banking site, the key used to create the encryption has to be negotiated 'in the clear'. What would be the point of setting up an encrypted secure channel when the key had previously been sent across the network for anyone to see? Of course the key isn't sent across the network and the following document describes *very simply* how this key negotiation is done without the actual key being sent over the network, where of course it could be seen by anybody running Ethereal or similar network sniffing software.

The method shown below is based on the Diffie-Hellman Key Exchange which was first published in 1976. See NOTE at the end of this document.

The exchange that can be transmitted openly across the network is shown in bold.

Client tells Server the starting number. This is a prime number generated at random

Client Tells Server: **STARTNUM = 5**

Client then picks another random number that is not disclosed.

Client Secret Number: CLI_SECNUM = 6

Client does the following maths: $STARTNUM^{CLI_SECNUM} = 15625$

Client tells the Server: **CLI_PUBNUM = 15625**

Server picks a random number that is not disclosed.

Server Secret Number: SVR_SECNUM = 3

Server does the following maths: $STARTNUM^{SVR_SECNUM} = 125$

Server tells the Client: **SVR_PUBNUM = 125**

Client does the following maths:
 $SVR_PUBNUM^{CLI_SECNUM} = 3814697265625$

Server does the following maths:
 $CLI_PUBNUM^{SVR_SECNUM} = 3814697265625$

The Client and Server now have a number (3814697265625), a key, that can be used to encrypt any further transmissions between them.

So the key that was calculated in public is secret and known only to the Client and Server. This works because exponential maths is not affected by the order in which the multiplications are done (power associative), and it is virtually impossible for the 3rd Party

using the data available to it (5, 15625 and 125) to calculate the secret key as it is missing the equally important data of 6 and 3 (the secret numbers).

In reality though the real strength of this method lies in the size of the numbers that are used. Our example uses very small numbers so that the maths are easily checked, a real world example would use numbers that were into the billions, depending on the bit length of the encryption used. The typical length of encryption used for this type of key exchange would be at least 512 bits.

NOTE: In actual fact the Diffie-Hellman Key Exchange works as shown above with the addition of a second prime number being used in conjunction with the STARTNUM, this second number is a primitive root modulo. I have avoided this in the calculations shown above for the sake of clarity.

Confidential