Diffie - Hellman Key Exchange

Networking Assignment CEC11

SATWIK	2018UC01647
RAJNISH	2018UC01653
SACHIN	2018UC01663
RAHUL	2018UC01665
RITVIK	2018UC01667
GEETANSH	2018UC01668

WHAT IS DIFFIE - HELLMAN KEY EXCHANGE

Diffie–Hellman key exchange is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman

DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography.

Where it is used

Public key encryption schemes based on the Diffie–Hellman key exchange have been proposed. The first such scheme is the ElGamal encryption. A more modern variant is the Integrated Encryption Scheme.

Diffie-Hellman is currently used in many protocols, namely:

- **Secure Sockets Layer (SSL)**
- **Transport Layer Security (TLS)**

- **SecureShell (SSH)**
- **Internet Protocol Security (IPSec) – Public Key Infrastructure (PKI)**

How it was different

Traditionally, secure encrypted communication between two parties required that they first exchange keys by some secure physical means, such as paper key lists transported by a trusted courier. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Diffie–Hellman is used to secure a variety of Internet services. However, research published in October 2015 suggests that the parameters in use for many DH Internet applications at that time are not strong enough to prevent compromise by very well-funded attackers, such as the security services of large governments

The method was followed shortly afterwards by RSA, an implementation of public-key cryptography using asymmetric algorithms.

Working and Algorithm

GENERAL IDEA OF WORKING

Diffie–Hellman key exchange establishes a shared secret between two parties that can be used for secret communication for exchanging data over a public network. An analogy illustrates the concept of public key exchange by using colors instead of very large numbers:

Lets see this example with actual math

he simplest and the original implementation[2] of the protocol uses the multiplicative group of integers modulo *p*, where *p* is prime, and *g* is a primitive root modulo *p*. These two values are chosen in this way to ensure that the resulting shared secret can take on any value from 1 to *p*–1. Here is an example of the protocol, with non-secret values in blue, and secret values in red.

1. **Alice and Bob publicly agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).**
2. **Alice chooses a secret integer $a = 4$, then sends Bob $A = g^a \bmod p$**
   - **$A = 5^4 \bmod 23 = 4$**
1. **Bob chooses a secret integer $b = 3$, then sends Alice $B = g^b \bmod p$**
   - **$B = 5^3 \bmod 23 = 10$**
2. **Alice computes $s = B^a \bmod p$**

- $s = 10^4 \bmod 23 = 18$

3. Bob computes $s = A^b \bmod p$
   - $s = 4^3 \bmod 23 = 18$
4. Alice and Bob now share a secret (the number 18).

Both Alice and Bob have arrived at the same values because under mod p,

$$A^b \bmod p = g^{ab} \bmod p = g^{ba} \bmod p = B^a \bmod p$$

Only *a*, *b*, and ($g^{ab} \bmod p = g^{ba} \bmod p$) are kept secret. All the other values – *p*, *g*, $g^a \bmod p$, and $g^b \bmod p$ – are sent in the clear. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel.

Of course, much larger values of *a*, *b*, and *p* would be needed to make this example secure, since there are only 23 possible results of *n* mod 23. However, if *p* is a prime of at least 600 digits, then even the fastest modern computers cannot find *a* given only *g*, *p* and $g^a \bmod p$. Such a problem is called the **discrete logarithm problem.**[3] The computation of $g^a \bmod p$ is known as **modular exponentiation** and can be done efficiently even for large numbers. Note that *g* need not be large at all, and in practice is usually a small integer (like 2, 3, ...).

s is the shared secret key and it is known to both Alice and Bob.

If any third party gets access to the messages a,b,s will remain hidden from it as these are never share between the persons communicating.

WHAT IS EPHEMERAL DIFFIE HELLMAN

Ephemeral Diffie-Hellman uses temporary, public keys. Each instance or run of the protocol uses a different public key. The authenticity of the server's temporary key can be verified by checking the signature on the key. Because the public keys are temporary, a compromise of the server's long term signing key does not jeopardize the privacy of past sessions. This is known as Perfect Forward Secrecy (PFS).

Advantages

- **Simple, and Elegant**
- **Sender, receiver don't need any prior knowledge of each other**
- **Works on insecure paths**
- **Secret Key is never shared**
- **Computationally very hard to crack the code**
- **There is, however, an advantage of DH over RSA for generating ephemeral keys: producing a new DH key pair is extremely fast (provided that some "DH parameters", i.e. the group into which DH is computed, are reused, which does not entail extra risks, as far as we know). This is not a really strong issue for big servers, because a very busy SSL server could**

generate a new "ephemeral" RSA key pair every ten seconds for a very small fraction of his computing power, and keep it in RAM only, and for only ten seconds, which would be enough.

Disadvantages

The biggest weakness is that, alone, it doesn’t establish the identity of the other party, making it susceptible to a man-in-the-middle attack.
For example, A wishes to communicate securely with B, and use DH to exchange keys.
Unbeknownst to either of them, M has rerouted their traffic to himself.
Since DH doesn’t establish identity, when A thinks he is negotiating a key with B, he is actually negotiating with M, and likewise(separately) for B, each then believes he is communicating securely with the other, when in fact, they each have a uniquely keyed secure connection with the eavesdropper who reads their messages, then re-encrypts them and sends them on, or reads them and sends different messages instead.

Can not be used for signing digital signatures