The Diffie–Hellman Key Exchange Algorithm

Dr. Kusan Biswas[*]
kusan.biswas@dseu.ac.in

September 5, 2025

The Diffie–Hellman Key Exchange algorithm was invented by Whitefield Diffie and Martin Hellman in 1976. It enables two parties to calculate a single secret key without first exchanging (*i.e.*, transmittig) anything secret. In other words, the two parties first exchange with each other some *public information*, and since these information are public, these information can be transmitted over an insecure channel. Using these public information, the two parties calculate two values, which turn out to be the *same*!. This shared secret value can not be computed any third party even with the knowledge of the public information the two parties exchanged over an insecure channel! This shared secret value can be used as the secret key of a Private Key Cryptographic algorithm such as AES.

Imagine two friends – Naveen and Praveen, wish to secretly transfer a big amount of data to each other. They have two options. One option is to use a public key cryptography algorithm such as the RSA. However, the RSA encryption and decryption processes are slow to run when the amount of data that is to be transmitted is large. The second option is to use a symmetric key cryptography algorithm such as AES. However, in the symmetric key cryptography there is only one key which must be used by both Naveen and Praveen. Either Naveen or Praveen must first generate the key. Let's say Naveen generates the key. Praveen also should have this key. How would Naveen share the key to Praveen? Naveen can't send the key over the network because the network is not secure yet. Naveen can visit Praveen and give the key to Praveen in person. This is not always feasible, because the Naveen and Praveen may be a very long distance apart. This is where the DHKE algorithm come into picture. The DHKE algorithm works as follows.

The Diffie–Hellman Key Exchange Algorithm

1. Creation of public parameters
   - Naveen chooses a prime number p and a base g
   - Naveen sends both p and g to Praveen
   - p and g can be sent to Praveen over an insecure channel, because these are public parameters. A malicious third party (i.e., a hacker) may intercept these parameters, but it is not a problem.
2. Private computation by Naveen and Praveen
   - Naveen chooses an integer a and keeps it secret
   - Praveen chooses an integer b and keeps it secret
   - Naveen computes $A = g^a \, mod \, p$
   - Praveen computer $B = g^b \, mod \, p$
3. Public exchange of values
   - Naveen sends A to Praveen over the insecure channel. No problem if any hacker intercepts it.
   - Praveen sends B to Naveen over the insecure channel. A hacker may read it, but no problem.
4. Further private computation
   - Naveen computes $A' = B^a \, mod \, p$
   - Praveen computes $B' = A^b \, mod \, p$
   - $A' = B'$ is the shared secret key! Mathematical properties of modular arithmetic ensure that the values of A' and B' computed this way are the *same*! This way, two parties – Naveen and Praveen *magically* compute the same secret key by exchanging a few public information. Once the two parties have the same secret key, they can exchange messages encrypted with symmetric key cryptographic algorithms such as AES and DES.

[*] Typeset in markdown backed by LaTeX! Hosted at https://github.com/kusanvision/NOTES

An Example of Diffie–Hellman Key Exchange Algorithm

1. Creation of public parameters
    - Naveen chooses a prime number $p = 23$ and a base $g = 5$
    - Naveen sends both $p = 23$ and $g = 5$ to Praveen
2. Private computation by Naveen and Praveen
    - Naveen chooses an integer $a = 6$ and keeps it secret
    - Praveen chooses an integer $b = 15$ and keeps it secret
    - Naveen computes $A = g^a \, mod \, p$

$$\begin{aligned} A &= g^a \, mod \, p \\ &= 5^6 \, mod \, 23 \\ &= 8 \end{aligned}$$

    - Praveen computer $B = g^b \, mod \, p$

$$\begin{aligned} A &= g^a \, mod \, p \\ &= 5^{15} \, mod \, 23 \\ &= 19 \end{aligned}$$

3. Public exchange of values
    - Naveen sends $A = 8$ to Praveen
    - Praveen sends $B = 19$ to Naveen
4. Further private computation
    - Naveen computes $A' = B^a \, mod \, p$

$$\begin{aligned} A' &= B^a \, mod \, p \\ &= 19^6 \, mod \, 23 \\ &= 2 \end{aligned}$$

    - Praveen computes $B' = A^b \, mod \, p$

$$\begin{aligned} B' &= A^b \, mod \, p \\ &= 8^{15} \, mod \, 23 \\ &= 2 \end{aligned}$$

    - $A' = B' = 2$ is the shared secret key! This way, two parties – Naveen and Praveen *magically* compute the same secret key by exchanging a few public information. Now Naveen and Praveen have the same secret key, therefore they can exchange messages encrypted with symmetric key cryptographic algorithms such as AES and DES.

Exercises

1. Take public parameters $p = 7$, $g = 3$. First party's secret $a = 2$ and second party's secret $b = 5$. Find out the secret key that both parties compute.
2. Take $p = 17$, $g = 5$. Alice's secret value $a = 4$ and Bob's secret value $b = 6$. Find out the secret key that both Alice and Bob compute[1]

[1] Typeset in markdown backed by LaTeX! Hosted at https://github.com/kusanvision/NOTES