Intercambio de claves de Diffie y Hellman

Adrián Racero Serrano
Juan Manuel Cardeñosa Borrego

Índice

1 Introducción

El algoritmo Diffie-Hellman debe su nombre a sus creadores Whitfield Diffie y Martin Hellman. Creado en 1976, es uno de los protocolos de intercambio de claves más antiguos que todavía se siguen usando en la actualidad. Sus creadores fueron galardonados con el premio A.M. Turing 2015 por este trabajo, con el que revolucionaron por completo la seguridad informática.

Este algoritmo permite a dos usuarios cualesquiera intercambiar, de forma confidencial, una clave secreta K (o de sesión) para posteriormente cifrar de forma simétrica los mensajes entre ellos dos.

2 Algoritmo

El funcionamiento de este algoritmo es más sencillo de lo que parece y se usa frecuentemente en protocolos y aplicaciones de encriptado de datos, como SSL (Secure Sockets Layer), SSH (Secure Shell) o VPN (Virtual Private Network). Este algoritmo permite que dos entidades (A y B) puedan generar una clave K_{AB} de forma simultánea, y sin enviarla por el canal de comunicaciones.

1) Para ello, A y B necesitan establecer y compartir valores comunes, como un valor q primo y una raíz primitiva α de q.

Para todo primo q existe un elemento $\alpha \in (\mathbb{Z}/q\mathbb{Z})^\times$ con $\text{ord}(\alpha) = q - 1 = \phi(q) = \#(\mathbb{Z}/q\mathbb{Z})^\times$.

$(\mathbb{Z}/q\mathbb{Z})^\times$ es el grupo multiplicativo de las clases de equivalencia módulo q, es decir, que no tienen ningún factor primo en común con q:

$$(\mathbb{Z}/q\mathbb{Z})^\times = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}.$$

Este α se llama una **raíz primitiva** módulo q. Ejemplo:

$$q = 3,\ \alpha = 2;\ 2^1 \ mod\ 3 = 1,\ 2^2 \ mod\ 3 = 2$$

2) Tanto A como B generan sus claves privadas ($X_{A/B}$) y públicas ($Y_{A/B}$) teniendo en cuenta que: $Y_A \equiv \alpha^{XA} \pmod q$, donde $0 \leq X_A \leq (q-1)$, X_A es el logaritmo discreto de Y_A, y se representa $dlog_{\alpha,q}\ (Y_A)$. Por tanto, la efectividad del algoritmo depende de la dificultad de computar logaritmos discretos.

3) Tanto A como B comparten sus respectivas claves públicas (Y_A/Y_B).

4) Tanto A como B generan la clave de sesión teniendo en cuenta:

$$K_{AB} = (Y_B)^{X_A} \mod q \rightarrow K_{AB} = (\alpha^{X_B})^{X_A} \mod q$$
$$K_{AB} = (Y_A)^{X_B} \mod q \rightarrow K_{AB} = (\alpha^{X_A})^{X_B} \mod q$$

Por tanto:

$$K_{AB} = \alpha^{X_B X_A} \mod q = \alpha^{X_A X_B}$$

3 Código

La clase **DiffieHellman**:

```
1  from funciones_auxiliares import *
2
3  class DiffieHellman :
4
5      def __init__(self, q, alfa):
6
7          if not(es_primo(q)):
8              print("ERROR: q no es primo")
9              exit(-1)
10
11         if not(es_raiz_primitiva(alfa, q)):
12             print("ERROR: alfa no es raiz primitiva de
                   q")
13             exit(-1)
14
15         self.__q__ = q
16         self.__alfa__ = alfa
17
18         self.__generarClaves__()
19
20     def __generarClaves__(self):
21         # X < q
22         self.__clavePrivada__ = numero_aleatorio(self.
               __q__)
23         # Y = alfa ^X mod q
24         self.__clavePublica__ = ( self.__alfa__ **
               self.__clavePrivada__ ) % self.__q__
25
26     def compartir_clave_publica(self):
27         return self.__clavePublica__
28
29     def generar_clave_secreta(self, clavePublicaB):
30         # K = Yb ^Xa mod q
31         self.__claveSecreta__ = ( clavePublicaB **
               self.__clavePrivada__ ) % self.__q__
32
33     # Este metodo solo se utiliza para comprobar
34     # que ambas claves secretas son iguales.
```

```
35      # Las claves secretas no se deben compartir.
36      def compartir_clave_secreta(self):
37          return self.__claveSecreta__
38
39      # Cifrar un mensaje usando XOR y la clave secreta
40      def cifrar_xor(self, mensaje):
41          cifrado = [char ^ self.__claveSecreta__ for
                char in mensaje.encode()]
42          return bytes(cifrado)
43
44      # Descifrar un mensaje usando XOR y la clave
            secreta
45      def descifrar_xor(self, cifrado):
46          mensaje = ''.join([chr(char ^ self.
                __claveSecreta__) for char in cifrado])
47          return mensaje
```

Ejecucion del algoritmo:

```
1  from algoritmo import *
2
3  print("------ INICIALIZAR VALORES ------")
4  print("q = 153")
5  print("alfa = 3")
6  print()
7  DH1 = DiffieHellman(353,3)
8  DH2 = DiffieHellman(353,3)
9
10 print("------ GENERAR CLAVE SECRETA ------")
11 DH1.generar_clave_secreta(DH2.compartir_clave_publica
       ())
12 print("Clave de DH1: " + str(DH1.
       compartir_clave_secreta()))
13 DH2.generar_clave_secreta(DH1.compartir_clave_publica
       ())
14 print("Clave de DH2: " + str(DH2.
       compartir_clave_secreta()))
15 print()
16
17 print("------- CIFRADO/DESCIFRADO -------")
18 mensaje = "Diffie Hellman"
19 cifrado = DH1.cifrar_xor(mensaje)
20 print("DH1 cifrando...")
21 print(mensaje + " -> " + str(cifrado))
22 mensaje_descifrado = DH2.descifrar_xor(cifrado)
```

```
23 print("DH2 descifrando...")
24 print(str(cifrado) + " -> " + mensaje_descifrado)
```

Salida de la ejecución:

```
------ INICIALIZAR VALORES ------
q = 153
alfa = 3

------ GENERAR CLAVE SECRETA ------
Clave de DH1: 231
Clave de DH2: 231

------- CIFRADO/DESCIFRADO -------
DH1 cifrando...
Diffie Hellman -> b'\xa3\x8e\x81\x81\x8e\x82\xc7\xaf
\x82\x8b\x8b\x8a\x86\x89'
DH2 descifrando...
b'\xa3\x8e\x81\x81\x8e\x82\xc7\xaf
\x82\x8b\x8b\x8a\x86\x89' -> Diffie Hellman
```

4 Autenticación

En el mundo real, el intercambio de claves Diffie-Hellman rara vez se utiliza por sí solo. La razón principal detrás de esto es que no proporciona autenticación, lo que deja a los usuarios vulnerables a los ataques de intermediarios.

Estos ataques pueden tener lugar cuando el intercambio de claves Diffie-Hellman se implementa por sí mismo, porque no tiene forma de verificar si la otra parte en una conexión es realmente quien dice ser. Sin ninguna forma de autenticación, los usuarios pueden conectarse con atacantes cuando creen que se están comunicando con una parte de confianza.

Por esta razón, el intercambio de claves Diffie-Hellman generalmente se implementa junto con algunos medios de autenticación. Esto a menudo implica el uso de certificados digitales y un algoritmo de clave pública, como RSA, para verificar la identidad de cada parte.

5 Seguridad

El intercambio de claves de Diffie-Hellman es considerado bastante seguro debido a la complejidad computacional asociada a los logaritmos discretos. Este protocolo de intercambio de claves utiliza operaciones expo-

nenciales y módulo sobre números primos grandes para generar claves compartidas sin la necesidad de transmitir información sensible entre las partes. La esencia de su seguridad radica en la dificultad de calcular el logaritmo discreto, es decir, encontrar la clave privada a partir de la clave pública en un entorno de campo finito.

La complejidad del logaritmo discreto impide eficientemente la resolución del problema mediante métodos de fuerza bruta o algoritmos conocidos. Aunque los avances en el poder de cómputo han permitido el desarrollo de métodos más sofisticados, la resistencia del Diffie-Hellman persiste debido a la imposibilidad de realizar cálculos inversos de manera eficiente en el entorno criptográfico.

Es importante destacar que la seguridad del Diffie-Hellman se basa en la dificultad de calcular logaritmos discretos, lo que implica que no hay un algoritmo eficiente para obtener la clave privada a partir de la clave pública. Sin embargo, a medida que la computación cuántica avanza, se plantean preocupaciones sobre la vulnerabilidad potencial del Diffie-Hellman a ciertos algoritmos cuánticos, lo que motiva la investigación de técnicas criptográficas postcuánticas.

Además, en la implementación práctica del Diffie-Hellman, es común utilizar medidas adicionales de seguridad, como la autenticación mediante certificados digitales y algoritmos de clave pública, para garantizar que las partes involucradas en el intercambio sean quienes dicen ser. Estas medidas adicionales abordan las limitaciones del Diffie-Hellman relacionadas con la autenticación y fortalecen aún más la seguridad del protocolo en entornos del mundo real. En resumen, la seguridad del intercambio de claves de Diffie-Hellman se basa en la complejidad del logaritmo discreto y se complementa con prácticas de implementación que abordan sus limitaciones en términos de autenticación.

6 Variaciones del intercambio de claves Diffie-Hellman

El intercambio de claves Diffie-Hellman se puede implementar de varias formas diferentes, y también ha proporcionado la base para varios otros algoritmos. Algunas de estas implementaciones proporcionan autorización, mientras que otras tienen varias características criptográficas, como el perfecto secreto hacia adelante.

6.1 Curva elíptica Diffie-Hellman

Curva elíptica Diffie-Hellman aprovecha la estructura algebraica de las curvas elípticas para permitir que sus implementaciones logren un nivel similar de seguridad con un tamaño de clave más pequeño. Una clave de curva elíptica de 224 bits proporciona el mismo nivel de seguridad que una clave RSA de 2048 bits. Esto puede hacer que los intercambios sean más eficientes

y reducir los requisitos de almacenamiento.

Aparte de la longitud de clave más pequeña y el hecho de que se basa en las propiedades de las curvas elípticas, Diffie-Hellman de curva elíptica opera de manera similar al intercambio de claves Diffie-Hellman estándar.

6.2 TLS

TLS, que es un protocolo que se utiliza para proteger gran parte de Internet, puede utilizar el intercambio Diffie-Hellman de tres formas diferentes: anónima, estática y efímera. En la práctica, solo se debe implementar Diffie-Hellman efímero, porque las otras opciones tienen problemas de seguridad.

- **Diffie-Hellman anónimo:** esta versión del intercambio de claves Diffie-Hellman no utiliza ninguna autenticación, lo que la deja vulnerable a los ataques de intermediarios. No debe usarse ni implementarse.
- **Static Diffie-Hellman:** Diffie-Hellman estático utiliza certificados para autenticar el servidor. No autentica al cliente de forma predeterminada, ni proporciona secreto hacia adelante.
- **Diffie-Hellman efímero:** se considera la implementación más segura porque proporciona un secreto directo perfecto. Generalmente se combina con un algoritmo como DSA o RSA para autenticar a una o ambas partes en la conexión. Efímero Diffie-Hellman utiliza diferentes pares de claves cada vez que se ejecuta el protocolo. Esto le da a la conexión un perfecto secreto hacia adelante, porque incluso si una clave se ve comprometida en el futuro, no se puede usar para descifrar todos los mensajes pasados.

7 Conclusión

En conclusión, el intercambio de claves de Diffie-Hellman ha demostrado ser una piedra angular en la construcción de comunicaciones seguras en entornos digitales. A lo largo de décadas, este protocolo ha ofrecido una solución eficaz para el desafío fundamental de establecer claves compartidas en un canal inseguro.

La gran característica del intercambio de claves de Diffie-Hellman radica en su capacidad para garantizar la confidencialidad de las claves compartidas, incluso en un escenario donde un atacante puede interceptar la comunicación. Su enfoque basado en la complejidad computacional de los logaritmos discretos ha resistido la prueba del tiempo, proporcionando una base sólida para numerosos protocolos de seguridad.

Sin embargo, es crucial reconocer las limitaciones inherentes del Diffie-Hellman, especialmente en lo que respecta a la autenticación. En el mundo

real, su implementación a menudo se combina con certificados digitales y algoritmos de clave pública para abordar la falta de verificación de la identidad de las partes involucradas. Esta adaptación ha permitido que el Diffie-Hellman se integre con éxito en protocolos como TLS/SSL, proporcionando una seguridad más completa y autenticada.

A medida que la ciberseguridad evoluciona, el intercambio de claves de Diffie-Hellman enfrentará desafíos continuos, desde la amenaza potencial de la computación cuántica hasta la necesidad constante de mejorar los protocolos de autenticación. Sin embargo, su impacto duradero en la seguridad de la información subraya su importancia como un pilar confiable en la construcción de sistemas de comunicación seguros y confiables. En un panorama digital en constante cambio, el Diffie-Hellman sigue siendo un contribuyente fundamental a la garantía de la privacidad y la integridad de las comunicaciones en línea.