Práctica 5
Facultad de Ciencias
Criptografía y Seguridad

José de Jesús Galaviz Casas
galaviz@ciencias.unam.mx

Edgar Omar Arroyo Munguía
omar.am@ciencias.unam.mx

Luis Fernando Yang Fong Baeza
fernandofong@ciencias.unam.mx

29 de Enero 2020

1. Intercambio de llaves de Diffie-Hellman

El intercambio de llaves de Diffie-Hellman, se utiliza para intercambiar información de manera eficiente, asegurando que un atacante, no pueda obtener la información suficiente para obtener la información del emisor enviada hacia el receptor.

Para ésta práctica se explicará el funcionamiento del algoritmo y el funcionamiento de la implementación para poder montarlo en cualquier servidor, para empezar una comunicación segura.

2. Algoritmo

Como siempre, digamos que Alice y Bob desean enviarse un mensaje, de manera que acuerdan de manera pública, fijar un primo p y una raíz primitiva módulo p, supongamos que es g. Supongamos que Carl, es un atacante que desea obtener toda la información intercambiada entre Alice y Bob, entonces Carl conoce a p y g.

Posteriormente, Alice dese comunicarse o mandarle un mensaje a Bob, entonces Alice escoge un entero a $(mod\ p)$ desconocido para todos, pero calcula $A \equiv g^a\ mod\ p$ y se lo manda a Bob, Bob procede de manera exactamente análoga pero para un número $B \equiv g^b\ mod\ p$, entonces le manda a Alice, el número B, Carl conoce a A y a B.

Por último, Alice recibe a A pero va a calcular $s = B^a\ mod\ p$, que recordemos que $B = g^b$, entonces Alice tiene a $g^{ba}\ mod\ p$, de manera análoga Bob puede obtener el mismo número pero no hay manera de que Carl obtenga este número s, puesto que desconoce ambos elementos a y b, entonces Carl puede ver a $p, g, A\ y\ B$, pero no puede conocer s, entonces de esta manera, vuelve seguro el intercambio de llaves, siendo a, la llave de Alice y b la llave de Bob, aplicando logaritmo discreto, que es claro que existe, por la formación original o incluso, utilizar a s como llave.

3. Implementación

Para fines prácticos de esta práctica no se probará con primos muy grandes puesto que calcular una raíz primitiva de un primo como el generador de la práctica pasada puede llegar a ser bastante tardado o problemático, puesto que tendríamos que verificar sobre $p-1$ elementos, asegurándonos

que ningún elemento $1 \leq k < p - 1$, cumple que $g^k \equiv 1 \ mod \ p$.

En la implementación tenemos una clase llamada *Participant*, que tiene los parámetros públicos, p y g y un participante con quien establece su comunicación y además, calcula su número secreto s. El funcionamiento de la práctica está basado en un problema productor/consumidor. Se genera el número secreto y se calcula g^s, en cuanto el participante almacenado como parámetro, haya visto g^s, se debe calcular un nuevo número secreto s pero debí haber visto antes el número del participante almacenado para guardar siempre la misma palabra clave.

De manera que quienes utilizan tal cual a Alice y Bob, son las pruebas unitarias y verifican que al final de que ambos manden a llamar su produce, el número sea exactamente el mismo.