Miramontes Sarabia Luis Enrique

Septiembre 26, 2019

ElGamal

ElGamal es un algoritmo propuesto por Taher Elgamal. Es una implementación del concepto de criptosistemas de llave asimétrica y está basado en el esquema de intercambio de llaves inventado por Diffie y Hellman. Su seguridad radica en la dificultad de calcular logaritmos discretos sobre campos finitos.

El algoritmo funciona de la siguiente manera; Supongamos que se tiene Alice y Bob que quieren compartir un secreto Kab, donde A tiene un secreto Xa y B tiene un secreto Xb. Se seleccionará un número primo P de gran longitud y un $\boldsymbol{\alpha}$ que será un elemento primitivo modulo p, ambos conocidos. A calculará Ya = $\boldsymbol{\alpha^{xa} \bmod p}$ y enviará Ya. Similarmente B calculará Yb = $\boldsymbol{\alpha^{xb} \bmod p}$ y enviará Yb. Entonces ahora el secreto Kab se calcula de la siguiente manera:

$$\text{Kab} = \boldsymbol{\alpha^{xaxb} \bmod p}$$
$$\boldsymbol{= Ya^{xb} \bmod p}$$
$$\boldsymbol{= Yb^{xa} \bmod p}$$

Por consiguiente, A y B son capaces de calcular Kab, pero para un intruso calcularlo es una tarea difícil. En los sistemas criptográficos basados en logaritmos discretos p debe de elegirse tal que p-1 tenga por lo menos un factor primo de gran tamaño. Si p-1 tiene solamente factores primos de pequeño tamaño el cálculo de logaritmos discretos se verá ampliamente simplificado.

Ahora supongamos que A quiere enviar a B un mensaje m, donde m es igual o mayor a cero y menor a p-1. Primero A selecciona un número k uniforme entre 0 and p-1, k servirá como el secreto xa en el esquema de distribución de llaves. Ahora A calcula la llave.

$$\text{K} = \boldsymbol{Yb^{k} \bmod p}$$

Donde $\boldsymbol{Yb = \alpha^{xb} \bmod p}$ se encuentra en un fichero público o es enviada a B. El ciphertext será la tupla **(C1, C2)** donde:

$$\boldsymbol{C1 = a^{k} \bmod p, \quad C2 = K*m \bmod p}$$

Para concluir ElGamal es un algoritmo basado completamente en el intercambio de llaves de Diffie-Hellman, a diferencia de RSA su complejidad no está basada en la complejidad de calcular los factores primos de n, sino en el cálculo de logaritmos discretos dentro de campos finitos. Igual tiene una complejidad alta y por lo que estuve buscando en internet no se ha roto como ha sucedido con los algoritmos tan viejos como lo es este.

Pese a esto RSA es más utilizado ya que es más rápido gracias a todas las simplificaciones vistas en clase y su fácil implementación a nivel de hardware. Que es lo que más se necesita en el área de la criptografía, seguridad a una velocidad razonable.