

## 8.4 Seguridad del Intercambio de Llaves de Diffie - Hellman

NOTA: un protocolo que usa la versión básica de DHKE no es seguro contra ataques activos.

Ataque de intermediario (MitM) → el adversario puede leer, insertar y modificar a voluntad  
↳ pasa cuando no hay autenticación  
↳ es necesario ser capaz de observar e interceptar mensajes

¿Qué pasa en caso de adversarios pasivos (sólo pueden leer mensajes)?

Objetivo de Óscar → calcular la llave de sesión  $k_{AB}$  compartida por Alice y Bob



Óscar



DHKE

Conoce  $\alpha$  y  $p$  ← públicos

Obtiene  $A = k_{\text{púb}, A}$  y  $B = k_{\text{púb}, B}$  ← espiando el canal en una ejecución

La pregunta aquí es: ¿Óscar es capaz de calcular  $k = \alpha^{ab}$  conociendo  $\alpha, p, A \equiv \alpha^a \pmod{p}$  y  $B \equiv \alpha^b \pmod{p}$ ?

PROBLEMA DE DIFFIE - HELLMAN  
(DHP)



Created with  
**Notewise**

## PROBLEMA DE DIFFIE - HELLMAN GENERALIZADO:

Dado  $G$  un grupo finito cíclico de orden  $n$ ,  $\alpha \in G$  un elemento generador y dos elementos  $A = \alpha^a$  y  $B = \alpha^b$ , encontrar  $\alpha^{ab} \in G$ .

Un enfoque general para resolver DHP: supongamos

- \* Estamos atacando DHP en  $(\mathbb{Z}/p)^*$ .
- \* Óscar conoce un método eficiente para calcular logaritmos discretos en  $(\mathbb{Z}/p)^*$ .

Con estas suposiciones, Óscar puede resolver DHP y obtener la llave  $k_{AB}$  en dos pasos:

① Calcular  $a = k_{pr,A}$  resolviendo el problema de logaritmo discreto  
 $a \equiv \log_\alpha A \pmod p$

② Calcular la llave de sesión  $k_{AB} \equiv B^a \pmod p$

¡PERO! Ya vimos que si  $p$  es suficientemente grande, calcular el problema de logaritmo discreto no es viable.

Para Óscar :

→ En este punto NO SE SABE si resolver DLP es la única forma de resolver DHP.

↳ En teoría PUEDE QUE EXISTA otra forma de resolver DHP SIN calcular el logaritmo discreto.

RSA  $\longleftrightarrow$  Factorización como mejor forma de romperlo



Sin embargo, se asume que resolver DLP EFICIENTEMENTE es la única forma de resolver DHP EFICIENTEMENTE.

Por lo tanto, para garantizar la seguridad de DHKE en la práctica, debemos asegurarnos de que el correspondiente DLP no se puede resolver.

→ Escoger  $p$  suficientemente grande para que el método de cálculo de índice no pueda calcular el DLP.

**Table 6.1** Bit lengths of public-key algorithms for different security levels

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

Requerimiento adicional (para prevenir ataques tipo Pohlig-Hellman):  
 $|(\mathbb{Z}/p)^*| = p-1$  no debe factorizarse solamente en factores primos pequeños



Cada subgrupo de orden alguno de los factores primos puede ser atacado usando el método baby-step giant-step o el método rho de Pollard

Por lo tanto, el menor factor primo de  $p-1$  debe tener al menos 160 bits de longitud para una seguridad de 80 bits, y al menos 256 bits de longitud para una seguridad de 128 bits.

