Trabajo Práctico N°3

Seguridad Informática

Alumno: Navall, Nicolás Uriel. N-1159/2.

2) a) Si tuviera que encriptar con dos claves k_1 y k_2 entonces escriptaria un mensaje m de la siguiente manera

$$t=DES_e(DES_e(m,k_1),k_2)$$

y desencriptariá como:

$$m=DES_d(DES_d(t,k_2),k_1)$$

b) El sistema es vulnerable a un ataque conocido como Meet-in-the-Middle

Supongamos que el atacante obtiene l pares de mensajes sin encriptar y su correspondiente encriptación: (p1,c1),...,(pl,cl). Puede demostrarse que para todo 1<=i<=l, DES_e(pi,k1)= DES_d(ci,k2). Luego, si tenemos las suficientes pares de mensajes encriptados y sin encriptar (si l>=(2*n)/m donde n es la longitud de la clave (56 bits) y m la longitud del mensaje) y encontramos dentro del espacio de claves disponible un par (k1',k2') los cuales para todo 1<=i<=l valga DES_e(pi,k1')= DES_d(ci,k2') entonces hay una gran probabilidad de que el par de claves encontradas sean las correctas, es decir (k1,k2)=(k1',k2').

En más detalle, el atacante mantiene dos listas L1 y L2 como se ve a continuación:

L1: Mensaje Original	Clave
$DES_e(k1_1, p_1)DES_e(k1_1, p_2)\dots DES_e(k1_1, p_l)$	$k1_1$
…	…
$DES_e(k1_{2^n}, p_1)DES_e(k1_{2^n}, p_2)\dots DES_e(k1_{2^n}, p_l)$	$k1_{2^n}$

L2: Mensaje Cifrado	Clave
$DES_d(k2_1, c_1)DES_d(k2_1, c_2)\dots DES_d(k2_1, c_l)$	$k2_1$
…	…
$DES_d(k2_{2^n}, c_1)DES_d(k2_{2^n}, c_2)\dots DES_d(k2_{2^n}, c_l)$	$k2_{2^n}$

El atacante ahora busca la listas L1 y L2 y revisa si hay alguna fila de L1 que sea igual a una fila en L2. De encontrar dos filas iguales, si se tiene l>=2, hay una gran probabilidad de que las claves encontradas sean las originales. Como cada lista tiene 2^n filas, y cada fila tiene l bloques de tamaño m bits cada uno, ademas de la clave correspondiente a dicha fila, cada fila tiene m*l+n bits, y cada lista ocupa 2^n*(m*l + n) bits, y como son dos necesitamos al menos 2^{n+1}*(m*l + n) bits de espacio de memoria para realizar el ataque descrito.

Por lo tanto, si el atacante tiene mucha memoria disponible, con el poder computacional de hoy en día, sería posible realizar el ataque descrito arriba.

3) Dada una clave *k* 3DES creará las claves k_1, k_2 y k_3 / $k= k_1+k_2+k_3$ (donde + es concatenación) y encriptará un mensaje t de la forma:

$$3DES_e(k,t)=DES_e(k_3,DES_d(k_2,DES_e(k_1,t)))$$

y descencriptará un mensaje t como:

$$3DES_d(k,t)=DES_d(k_1,DES_e(k_2,DES_d(k_3,t)))$$

Donde DES_e es el encriptador de DES y DES_d el desencriptador.

3DES no es vulnerable a ataques de tipo meet-in-the-middle por que este es equivalente a DES con una clave de 112 bits, lo cual implica miles de miles de terabytes de memoria necesaria para realizar el ataque, siendo este impracticable.

8) a) El ataque se basa en obtener de forma previa el tercer mensaje de una corrida del protocolo y extraer K de este (se debería romper la encriptación por lo que tomaría bastante tiempo, por eso no sería un inconveniente cuando esa corrida del protocolo está sucediendo). Luego cuando comienza una nueva corrida y B responde al Hello de A, C manda este mensaje de la corrida anterior a B, y B responderá con un mensaje encriptado con K, el cual C tiene y sigue la comunicación entre ellos normalmente.

b) Para corregirlo se podría mandar un fresco en el segundo mensaje

A->B: Hello, pbk(A)
B->A: aenc(pbk(A),N||CertB)
A->B: aenc(pbk(B),N-1||K||CertA)

Por lo cual A le manda su certificado a B en conjunto con la clave K que utilizaran para encriptar los mensajes de forma simétrica, además de enviar el fresco nuevamente para asegurarse que el paquete se esté generando en ese momento.

Luego continua la comunicación entre A y B de forma normal.