Lsningsfrslag till tenta i DD2495, Nätverkssäkerhet

2011-05-23

Del 1

1. Falskt.
2. Falskt.
3. Sant.
4. Falskt.
5. Sant.
6. Sant.
7. Sant.
8. Falskt.
9. Sant.
10. Sant.

Del 2

1. Vi tänker oss först att $M = 0$. Både DES och AES fungerar i stort sett så att M delas upp i delar och translateras. Nyckeln delas upp i olika delnycklar som sedan XORas ihop med delnycklarna som skapas med hjälp av huvudnyckeln. Dessa strängar skickas sedan in i en eller flera S-boxar. (En för AES och åtta för DES.) Även om både M och nyckeln skulle vara 0 kommer S-boxarna att returnera något annnat än 0. Motsvarande gäller om $K = 0$. S-boxarna är inte inverterbara p något enkelt sätt. Det innebär att det som skickas in i S-boxarna inte avslöjas av det som kommer ut. Även om $E(M, 0) = C$ s ger inte C någon enkel information om M .

2. a. De kommer från det grundläggande Client Hello-protokollet I SSL. (Se sid 479.) R_1 är ett nonce som genereras av A. R_2 är ett nonce som genereras av B. S är ett slumpvärde som genereras av A.

   b. Antag att $|R_1|$, $|R_2|$, $|S|$ r längden på strängarna. (Vi antar att alla strängar av samma typ har samma längd.) Låt $|K|$ vara längden på nycklarna. Stt $M_1 = 2^{|R_1|}2^{|R_2|}2^{|S|}$ och $M_2 = 2^{|K|}$. Då gäller $f : [M_1] \to [M_2]$. Först kan det verka rimligt att testa alla R_1, R_2, S om $M_1 < M_2$ och att testa alla K annars. Men om vi har $M_1 > M_2$ så gäller att vi har *kollisioner*. I medeltal kommer vi att hitta rätt nyckel efter $\frac{M_1}{M_2}$ försök.(Det är egentligen om vi testar nycklar med

återläggning. Det rimliga sättet att testa är förstås utan återläggning. Men vi kan anta att denna uppskattning är ungefär rätt även utan återläggning.)Om vi testar nycklar direkt så måste vi i medeltal testa $\frac{M_2}{2}$ nycklar. Om vi antar att tiden det tar att köra f relativt tiden det tar att testa varje nyckel är t så får vi att vi bör testa alla R_1, R_2, S om $\frac{M_1}{M_2}t < M_2$ och att vi bör testa alla K annars.

3. Vi använder lämpligen Ipsec i tunnelälge med ESP. Se beskrivningen p sid 425-426.

4. Vi kan anta att T kan avlyssna trafiken. Han kan fånga upp $E(A, N_1, K_{ab}), K_b)$ och $E((M, N_1), K_{ab})$. Han kan sedan sända båda dessa meddelanden upprepade gånger till B. I princip bör han kunna få 500 kr utbetalade varje gång.

   Ett lämpligt motmedel skulle kunna vara att lägga in en tidsstämpel i båda dessa meddelanden.

5. a. Att två certifikat blivit signerade med samma nyckel tyder bara på att två CA har samma privata nyckel. Detta kan vara ett större säkehetsproblem vid anarkimodellen men kanske ett mindre vid hierarkimodellen.

   b. Detta är ett stort problem. Det tyder på att två personer har samma publika och privata nycklar. Den ena personen kan då läsa krypterade meddelanden sända till den andra personen.

   c. Om två meddelanden har samma signaturer betyder det att vi fått en kollision i hashfunktionen som vi använder för signering. Det betyder att det finns en möjlighet att ett av certifikaten kan vara skapat av en pirat som har hittat kollisionen.

6. Autentisatorn mellan C och S innehåller klientens identitet och en tidsstämpel. Dessa delar är krypterade med en *sessionsnyckel*. En klient med en giltig autentiserare måste alltså ha en giltig sessionsnyckel. Dessa nycklar konstrueras av TGS och skickas till C och S. TGS legitimerar på så sätt att C är en betrodd användare. Eftersom autentiseraren innehåller en tidsstämpel så begränsas möjligheterna att göra en replay-attack.

7. a. Se sid 653.
   Innebär I stort att även om en angripare kan lyssna på en (krypterad) konversation och sedan komma över en av parternas långtidshemligheter så kan han inte dekryptera konversationen ändå.

   b. Se sid 658.

   c. Se sid 667.
   Här finns det utrymme för egna funderingar.