Computer Security

Reagan Shirk

October 8, 2020

Exercises

Exercise 1

Given a random key K of length m bits. K is used as a one time pad to encrypt P_1 (of length m bits) to obtain C_1. K is then also used to encrypt plaintext P_2 (of length m bits) to obtain C_2. This means that the key is reused. If the adversary is able to obtain C_1, C_2, and P_2 (known plaintext attack), explain how he can compute P_1:

You perform an $P_2 \oplus C_2$ to get the key, then the attacker can use K to decrypt C_1 and obtain P_1

Exercise 2

Random J Protocol-Designer has been told to design a scheme to prevent messages from being modified by an intruder. Random J decides to append to each message a hash of that message. Why doesn't this solve the problem? The attacker has the key/everything they would need to decrypt the message.

Anyone who hacks the system will be able to append a hash onto any message, there's no way to guarantee that the person appending the hash is legitimate. The attacker can decrypt the message, change it, reappend their hash and it looks legitimate.

Exercise 3

Alice needs to send n packets to Bob. Alice and Bob wish to use n different random one-time pads to encrypt the n packets. The encryption is simply bit-wise or XORing a packet and a one-time pad. Assume each packet is 128 bits in length. Describe how Alice and Bob can generate n different one-time pads by using a shared secret key S and cyrptographic hash function h.

The first one-time pad will be a hash function on S. After that, you will apply a bit-wise operation of your choosing to S to generate a new shared key. You can hash the new shared key for the new one-time pad. This can be done n times. You can also continue to hash the hashes, i.e. if $n = 3$ then $p_3 = h(S),\ p_2 = h(h(S)),\ p_1 = h(h(h(S)))$. $p_1,\ p_2,\ p_3$ can be used as one-time pads for encryption.

Exercise 4

Alice developed a MAC based on DES. Her algorithm is: For a given input message M, represent M as $M = (X_1||X_2||\cdots||X_3)$, where X_i is a 64-bit block and $||$ represents concatenation. Compute $\Delta(M) = X_1 \oplus X_2 \oplus \cdots \oplus X_m$. Then the MAC for M is computed as $CK(M) = E_k(\Delta(M))$ where E is DES encryption algorithm and K is the secret key. Unfortunately, the scheme is vulnerable, Describe an attack against it.

XORing plaintext blocks doesn't satisfy the strong collision free property. It's easy for an attacker to modify the original message m into a new m' such that both messages have the same MAC. For example:

$$
\begin{aligned}
M &= (100||001||111) \\
\Delta(M) &= 100 \oplus 001 \oplus 111 \\
CK(M) &= E_K(010) \\
M' &= (110||101||001) \\
\Delta(M') &= 110 \oplus 101 \oplus 001 = 010 \\
CK(M') &= E_K(\Delta(M')) = E_K(101) = CK(M)
\end{aligned}
$$

Exercise 5

If a good quality hash of a message produces a 256 bit output, how many messages would you need to try at random to have at least a 50% chance of generating the same hash output?

$$
\begin{aligned}
P(n,k) = 1 - \frac{n!}{(n-k)!n^k} &\approx 1 - e^{-k \times \frac{k-1}{2n}} \\
P(256,k) &= \frac{256!}{(256-k)!256^k} \\
k &= 2^{\frac{256}{2}} \\
&2^{128} \\
m &= 128
\end{aligned}
$$