Exercise 1 (10 points)

Assume that one *master key* k_{MK} is exchanged in a secure way (e.g., certificate based DHKE) between the involved parties. Afterwards, the session keys are regularly updated by use of *key derivation*. For this purpose, three different methods of key derivation are at our disposal:

Method 1:

$k_0 = k_{MK}$

$k_{i+1} = k_i + 1$, where $i \geq 0$

Method 2:

$k_0 = h(k_{MK})$

$k_{i+1} = h(k_i)$, where $i \geq 0$

Method 3:

$k_0 = h(k_{MK})$

$k_{i+1} = h(k_{MK} \,||\, i \,||\, k_i)$, where $i \geq 0$

Where $h(\bullet)$ is a cryptographically secure hash function.

Question 1

Assume Oscar obtains the *n*th session key (e.g., via brute-force). Which sessions can he now decrypt (depending on the chosen method)?

Question 2

Which method remains secure if the master key k_{MK} is compromised?

SOLUTION

Question 1.

Let us suppose that k_n is compromised.

Method 1. Oscar can decrypt the nth session, all previous, and all subsequent sessions.

Method 2. Oscar can decrypt the nth session and all subsequent sessions. For example, $k_{n+1} = h(k_n)$.

Method 3. Oscar can decrypt only the nth session. To compute k_{n+1} (subsequent session), Oscar needs k_{MK}, that is a secret. In order to compute k_{n-1} (previous session), Oscar needs to “invert” $h(\bullet)$ which is practically infeasible.

Question 2.

Method 1. Oscar can decrypt all sessions.

Method 2. Oscar can decrypt all sessions.

Method 3. Oscar can decrypt all sessions.