Crypto 2 Ü7

Ex 20.1

a.)

$h(k) = \lfloor M \cdot (k\phi - \lfloor k\phi \rfloor) \rfloor \geq 0$ $\quad \phi = \frac{1+\sqrt{5}}{2}$ // golden ratio

$(k\phi - \lfloor k\phi \rfloor) < 1$ $\quad M = 10.000$

$< M$

≤ 9999

$\Rightarrow 0 \leq h(k) \leq 9999$ It is not known if there is a k s.th. $h(k) = 0$ or $h(k) = 9999$

This specific type of hash-functions is called the multiplicative method (Fibonacci-hash)

Fibonacci: $F_n = F_{n-1} + F_{n-2}$, $F_0 = 0$

$\lim \frac{F_{n+1}}{F_n} = \phi = \frac{1+\sqrt{5}}{2}$; irrational

b.) finding collisions is very hard in general!

// use computer

$h(1) = 6180 = h(10947) = h(1 + 10946)$

$h(2) = 2360 = h(6767) = h(2 + 6765)$

$h(3) = 8541 = h(10949) = h(3 + 10946)$

$h(4) = 4721 = h(10950) = h(4 + \_)$

$h(5) = 901 = h(6770) = h(5 + \_)$

$\Rightarrow h(10946) = 0$ // 21st Fibonacci number

$h(6765) = 9999$ // 20st Fibonacci number

hash-values almost uniformly distributed

Ex 21.)

1.) $C_i = M_{i+1} \oplus E_k(C_{i-1}) \qquad i = 1, \dots, n-1$

2.) $MAC_k^{(a)} = E_k(C_{n-1})$

3.) $C_0 = M_1$

4.) $\hat{C}_i = E_k(\hat{C}_{i-1} \oplus M) \qquad i = 1, \dots, n$

5.) $\widehat{MAC}_k^{(a)} = E_k(\hat{C}_{n-1} \oplus M_n)$

6.) $\hat{C}_0 = 0$

a.) Show that the equivalence $MAC_k^{(a)} = \widehat{MAC}_k^{(a)}$ holds:

=> Induct over n

7.) $MAC_k^{(a)} = \widehat{MAC}_k^{(a)}$

proof:

$n = 1$: $MAC_k \overset{(4)(2)}{=} E_k(C_0') \overset{(3)}{=} E_k(M_1) \overset{(6)}{=} E_k(\hat{C}_0 \oplus M_1) \overset{(5)}{=} \widehat{MAC}_k^{(1)}$

$n = n+1$:

$$MAC^{(n+1)} \overset{(2)}{=} E_k(C_n') \overset{(1)}{=} E_k(M_{n+1} \oplus E(C_{n-1}))$$
$$\overset{(2)}{=} E_k(M_{n+1} \oplus MAC_k^{(n)})$$
$$\overset{(7)}{=} E_k(M_{n+1} \oplus \widehat{MAC}_k^{(n)})$$
$$\overset{(5)}{=} E_k(M_{n+1} \oplus E_k(\hat{C}_{n-1} \oplus M_n))$$
$$\overset{(4)}{=} E_k(M_{n+1} \oplus \hat{C}_n) \overset{(4)}{=} \hat{C}_{n+1} = \widehat{MAC}_k^{(n+1)}$$

□

Crypto 2 ÜA

06. DEZ. 2011

Ex 22.)

a.)

$A \rightarrow B \quad c = e(m \| h(k_2 \| m), k_1)$

B: • $d(c, k_1) = m \| h(k_2 \| m)$

• compute $h(k_2 \| m)$ with the shared key k_2

• verify $h(k_2 \| m)$ with the own computation

Background:

- 2 keys are used to separate encryption and message validation
- 2 keys can have different security levels
- encryption can be omitted
- If a part of the key is lost, the system is not entirely broken

b.)

$A \rightarrow B: \quad c = e(m \| h(s \| m) \| e(s, k_2), k_1)$

(k_1, L_1) } belongs to Bob
(k_2, L_2) }

B: • $d(c, L_1) = m \| h(s \| m) \| e(s, k_2)$

• $d(e(s, k_2), L_2) = s$

• compute $h(s \| m)$ with session key s

• verify $h(s \| m)$

* no authentication of A

$E \rightarrow B: \quad c$

E can easily impersonate Alice and Bob does not notice.

Alternative:

(VI) Alice starts session ("request session key")

$B \rightarrow A$: $c_1 = e(s, k_3)$ (k_3, L_3) belongs to A

A: $d(c_1, L_3) = s$

$A \rightarrow B$: $c_2 = e(m \| h(s \| m), k_4)$ (k_4, L_4) belongs to B

B: $d(c_2, L_4) = m \| h(s \| m)$

- compute $h(s \| m)$ with session keys
- verify $h(s \| m)$