RWTH AACHEN UNIVERSITY Lehrstuhl für Theoretische Informationstechnik

Exercise 6 in Advanced Methods of Cryptography - Proposed Solution -

Prof. Dr. Rudolf Mathar, Henning Maier, Markus Rothe
2014-11-28

Solution of Problem 17

a) With a block cipher $E_K(x)$ with block length k, the message is split into blocks m_i of length k each, $m = (m_0, \dots, m_{n-1})$. Take $m = (m_0)$ and $\hat{m} = (m_0, m_1, m_1)$, m_0, m_1 arbitrary. Then,

$$h(\hat{m}) = E_{m_0}(m_0) \oplus \underbrace{E_{m_0}(m_1) \oplus E_{m_0}(m_1)}_{=\mathbf{0}} = E_{m_0}(m_0) = h(m)\,.$$

Thus, h is neither second preimage resistant nor collision free.

Given $y \in \mathcal{Y}$, choose m_0. Then calculate

$$\begin{aligned} c &= E_{m_0}(m_0)\,, \\ m_1 &= D_{m_0}(c \oplus y)\,. \end{aligned}$$

It follows that

$$h(m_0, m_1) = E_{m_0}(m_0) \oplus E_{m_0}(D_{m_0}(c \oplus y)) = c \oplus c \oplus y = y\,.$$

Hence, h is *not* preimage resistant, either.

b) $\hat{h}$ replaces XOR ($\oplus$) by AND ($\odot$) and remains the same as h otherwise. Take $m = (m_1, m_1)$, with m_1 chosen arbitrarly. Then,

$$\hat{h} = E_{m_1}(m_1) \odot E_{m_1}(m_1) = E_{m_1}(m_1) = \hat{h}((m_1))\,.$$

$\hat{h}$ is neither second preimage resistant nor collision free.

Solution of Problem 18

a) Let $M = 10000$ and $\delta = \frac{1+\sqrt{5}}{2}$. Then,

$$h(k) = \underbrace{\left\lfloor\, \underbrace{M\,\underbrace{(k\delta - \lfloor k\delta \rfloor)}_{<1}}_{<10000} \,\right\rfloor}_{\leq 9999} \quad \Rightarrow \quad 0 \leq h(k) \leq 9999\,.$$

It is not yet known if there is a k such that $h(k) = 0$ or $h(k) = 9999$.

This specific type of hash function is called the *multiplicative method* (Fibonacci-hash). The hash values are almost uniformly distributed in $[0, 9999]$. Recall Fibonacci: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$. Then,

$$\lim_{n\to\infty} \frac{F_{n+1}}{F_n} = \frac{1+\sqrt{5}}{2},$$

which is an irrational number.

b) Finding collisions is very hard in general, so use your computer:

$$\begin{aligned}
h(1) &= 6180 = h(10947) = h(1 + 10946)\\
h(2) &= 2360 = h(6767) = h(2 + 6765)\\
h(3) &= 8541 = h(10949) = h(3 + 10946)\\
h(4) &= 4721 = h(10950) = h(4 + 10946)\\
h(5) &= 901 = h(6770) = h(5 + 6765)
\end{aligned}$$

Note that $h(10946) = 0$ (21st Fibonacci number) and $h(6765) = 9999$ (20th Fibonacci number).

Solution of Problem 19

$$\begin{aligned}
C_i &= M_{i+1} \oplus E_K(C_{i-1}),\ \ i = 1, \dots, n-1 &&(1)\\
\mathrm{MAC}_K^{(n)} &= E_K(C_{n-1}) &&(2)\\
C_0 &= M_1 &&(3)\\
\hat{C}_i &= E_K(\hat{C}_{i-1} \oplus M_i),\ \ i = 1, \dots, n-1 &&(4)\\
\widehat{\mathrm{MAC}}_K^{(n)} &= E_K(\hat{C}_{n-1} \oplus M_n) &&(5)\\
\hat{C}_0 &= 0 &&(6)
\end{aligned}$$

We show that the equivalency

$$\mathrm{MAC}_K^{(n)} = \widehat{\mathrm{MAC}}_K^{(n)} \qquad (7)$$

holds, by induction over n.

Proof. $n = 1$:

$$\mathrm{MAC}_K^{(1)} \overset{(2)}{=} E_K(C_0) \overset{(3)}{=} E_K(M_1) \overset{(6)}{=} E_K(\hat{C}_0 \oplus M_1) \overset{(5)}{=} \widehat{\mathrm{MAC}}_K^{(1)}$$

$n \to n+1$:

$$\begin{aligned}
\mathrm{MAC}_K^{(n+1)} &\overset{(2)}{=} E_K(C_n) \overset{(1)}{=} E_K(M_{n+1} \oplus E_K(C_{n-1}))\\
&\overset{(2)}{=} E_K\left(M_{n+1} \oplus \mathrm{MAC}_K^{(n)}\right)\\
&\overset{(7)}{=} E_K\left(M_{n+1} \oplus \widehat{\mathrm{MAC}}_K^{(n)}\right)\\
&\overset{(5)}{=} E_K\left(M_{n+1} \oplus E_K\left(\hat{C}_{n-1} \oplus M_n\right)\right)\\
&\overset{(4)}{=} E_K\left(M_{n+1} \oplus \hat{C}_n\right) \overset{(4)}{=} \hat{C}_{n+1} = \widehat{\mathrm{MAC}}_K^{(n+1)}
\end{aligned}$$

□

Solution of Problem 20

a) Alice sends to Bob:

$$c = e(m \parallel h(k_2 \parallel m), k_1)\,.$$

Bob validates as follows:

- $d(c, k_1) = m' \parallel h(k_2 \parallel m)$
- compute $h(k_2 \parallel m')$ with the known key k_2
- verify $h(k_2 \parallel m) = h(k_2 \parallel m')$

Background:

- two keys are use to separate encryption and validation
- e.g., two keys can have different security levels
- encryption can be omitted if the message is not secret, but integrity is still ensured
- if a part of the key is lost, then the system is not entirely broken

b) *Method (i)*: Let both, (K_1, L_1) and (K_2, L_2) belong to Bob.

First, Alice sends the following to Bob:

$$c = e(m \parallel h(s \parallel m) \parallel e(s, K_2), K_1)\,.$$

Then, Bob validates:

- $d(c, L_1) = m \parallel h(s \parallel m) \parallel e(s, K_2)$
- $d(e(s, K_2), L_2) = s$
- compute $h(s \parallel m')$ with session key s
- verify $h(s \parallel m) = h(s \parallel m')$.

Method (ii): Let (K_A, L_A) belong to Alice and (K_B, L_B) belong to Bob, then a possible protocol for message validation is.

- Bob sends to Alice: $c_1 = e(s, K_A)$
- Alice calculates: $d(c_1, L_A) = s$
- Alice then sends to Bob: $c_2 = e(m \parallel h(s \parallel m), K_B)$
- Bob calculates $d(c_2, L_B) = m \parallel h(s \parallel m)$, computes $h(s \parallel m')$ with session key s, and verifies that $h(s \parallel m) = h(s \parallel m')$.

There are many more protocols possible.

c) There is no authentication for Alice and Bob. Eve can easily impersonate Alice:

Eve sends the following c to Bob using Bob's public key:

$$c = e(\tilde{m} \parallel h(s \parallel \tilde{m}) \parallel e(s, K_2), K_1)$$

with her own forged message $\tilde{m}$ impersonating Alice. Bob does not know that this message is actually from Eve.

To counteract this attack, the message must be securely linked to Bob's identity. This demand can be accomplished by a signature. For instance:

$$c = e(m \parallel sig_A(m) \parallel h(s \parallel sig_A(m) \parallel e(s, K_2), K_1).$$