CS403 · Cryptography and Network Security

Assignment 2

Yubin Hu / 11712121

November 23, 2020

1

F is not a PRF.

Suppose we have the distinguisher D, D queries $F_{A,b}(0)$ which equals b. D can compute b with one query. Because b is known, and we know that D can compute Ax by every x. D can compute A by posing queries for the unit vectors. So F is not a PRF.

2

Assume the statement is false, which means there exist some PPT adversary A,

$$|Pr[PrivK^{CPA}_{A,\Pi}(n) = 1] - Pr[PrivK^{CPA}_{A,\tilde{\Pi}}(n) = 1]| > t(n)$$

For some non-negligible $t(n)$. We constructs a PPT distinguisher D contradicting the requirement from the definition of pseudo-random functions, so we have:

$$|Pr[D^{F_k(.)}(1^n)] - Pr[D^{f(.)}(1^n)]| > t(n)$$

We assume the working of the distinguisher D, which is given access to some oracle $O : \{0,1\}^n \to \{0,1\}^n$ and receives an input 1^n

- Run $A(1^n)$, and query $A(1^n)$'s oracle to the encryption function for i-th time with a message made up by l_i message blocks($m_1, m_2, .., m_{l_i}$), we do:
  - Choose a uniform initial value $ctr^* \in \{0,1\}^m$
  - Query O for $j = 1, .., l_i$ to obtain $y_j = O(ctr_i + j)$
  - Return the ciphertext blocks $< ctr_i, c_i, ..., c_{l_i} >=< ctr_i, y_i \oplus m_{b,1}, ..., y_{l*} \oplus m_{b,l^*} >$ to A.
- Once A output the messages m_0, m_1 consisting of l^* blocks $m_{0,1}, ..., m_{0,l*}, m_{1,1}, .., m_{1,l*}$ respectively, choose a uniform bit $b \in 0,1$, we do:
  - Choose a uniform initial value $ctr^* \in \{0,1\}^m$
  - Query O for $j = 1, .., l_i$ to obtain $y_j = O(ctr_i + j)$
  - Return the ciphertext blocks $< ctr_i, c_i, ..., c_{l_i} >=< ctr_i, y_i \oplus m_{b,1}, ..., y_{l*} \oplus m_{b,l^*} >$ to A.
- Answer queries to the encryption oracle as above, until A produces an output bit b'. Then output 1 if $b = b'$, and 0 otherwise.

We argue D is PPT hence D runs in polynomial time.

Note that D is essentially just the experiment $PrivK^{CPA}_{A,\Pi}$ or $PrivK^{CPA}_{A,\tilde{\Pi}}$ depending on witch oracle D is given. with the first step of key generation in the experiment being simulated by uniformly choosing $k \in \{0,1\}^n$ or $f \in Func_n$.

$$Pr_{k \leftarrow (0,1)^n}[D^{F_k(.)}(1^n) = 1] = Pr[PrivK^{CPA}_{A,\Pi}(n) = 1]$$
$$Pr_{k \leftarrow Func_n}[D^{F_k(.)}(1^n) = 1] = Pr[PrivK^{CPA}_{A,\tilde{\Pi}}(n) = 1]$$

3

3.1

This scheme has indistinguishable encryptions in the presence of an eavesdropper.

We prove that $F_k(0^n)$ is pseudorandom.

The scheme is not CPA-secure as encryption is deterministic.

3.2

The one-time pad is perfectly indistinguishable.

It has also indistinguishable encryption in the presence of an eavesdropper.

. It is not CPA-secure as encryption is deterministic.

3.3

This scheme does not even have indistinguishable encryptions in the presence of an eavesdropper.

It cannot be CPA-secure.

3.4

This scheme is CPA-secure.

It has indistinguishable encryptions in the presence of an eavesdropper

4

4.1

This is not CPA-secure. Follow this attacker A:

- The key k is chosen at random and fixed.
- A get 2 different messages $m_0, m_1 \in \{0,1\}^{2m}$. Then A interacts with the encryption oracle E_k and obtains two ciphertexts y_0, y_1
- A sends m_0, m_1 to the challenger and gets $c^* = E_k(m_b)$ for $b \leftarrow_R \{0,1\}$. A intercepts the first $2m$ bits of c^* as c'
- if c' is identical to the first $2m$ bits of y_0, A outputs $b = 0$; otherwise, $b = 1$

Therfore, A wins the game with probability 1.

4.2

This scheme is CPA-secure. Considering $y_1 = p_k(0^n \oplus r)$ as a random function.

5

We construct an adversary A for each of the MACs.

5.1

On input 1^n, A queries $(0^n 1^n)$ and gets $t = Mac_k(0^n 1^n) = F_k(0^n) \oplus F_k(1^n)$. Now A outputs $(1^n 0^n, t)$. This is a valid messages-tag pair as $Mac_K(1n0^n) = F_k(1^n) \oplus F_k(0^n) = F_k(0^n) \oplus F_k(1^n) = t$. So A wins with probability 1.

5.2

On input 1^n, A queries $m_0 = 0^n$, $m_1 = 0^{\frac{n}{2}}1^{\frac{n}{2}}$ and $m_2 = 1^n$. We donote the tags as t_0, t_1, t_2.

$$\begin{aligned} & t_0 \oplus t_1 \oplus t_2 \\ & \qquad = (F_k([1]||0^{\frac{n}{2}}) \oplus F_k([2]||0^{\frac{n}{2}}) \oplus F_k([1]||0^{\frac{n}{2}}) \oplus F_k([2]||1^{\frac{n}{2}}) \oplus F_k([1]||1^{\frac{n}{2}}) \oplus F_k([2]||1^{\frac{n}{2}})) \\ & \qquad = F_k([2]||0^{\frac{n}{2}}) \oplus F_k([1]||1^{\frac{n}{2}}) - F_k([1]||1^{\frac{n}{2}}) \oplus F_k([2]||0^{\frac{n}{2}}) \\ & \qquad = Mac_k(1^{\frac{n}{2}}0^{\frac{n}{2}}) \end{aligned} \tag{1}$$

Therfore, A outputs $(1^{\frac{n}{2}}0^{\frac{n}{2}}, t_0 \oplus t_1 \oplus t_2)$ and wins with probability 1.

5.3

Let $m \in 0, 1^{\frac{n}{2}}$ be an arbitary messages. Then A outputs $(m, ([1]_2||m))$. This is a valid message-tag pair as Mac_k could choose $r = [1]_2||m$ and output

$$t = (r, F_k(r) \oplus F_k([1]_2||m)) = (r, 0^n)$$

Therfore, A wins with probability 1.

6

For arbitary two messages $m_1, m_2 \in 0, 1^n$ and $m_1, m_2 \neq 0^n$. So we can query the oracle $t_1 = Mac_k(m_1||0^n), t2 = Mac_k(0^n||m_2)$. Then let c_1 be the first $\frac{n}{2}$ bits of t_1 and c_2 be the last $\frac{n}{2}$ bits of t_2. Then A can output $(m_1||m_2, c_1||c_2)$.

7

$\widetilde{H}$ is a collision resistant hash function. We prove by reduction.

Assume that $\widetilde{H}$ is not a collision resistant. Then there is a PPT adversary $\widetilde{A}$ such that:

$$Pr[Hash - col_{\widetilde{A},\widetilde{\Pi}}(n) = 1] \geq \frac{1}{q(n)}$$

We use $\widetilde{A}$ to constructs A as follows: On input s, A simulates $\widetilde{A}$. The latter will output x, x' eventually. Now A checks whether $H^8(x) = H^8(x')$ and $x \neq x'$. If this is not the case A will just output x and x'. Otherwise, A will checks whether $\widetilde{H}^8(x) = \widetilde{H}^8(x')$. If this is the case, then a collision was found and A outputs x and x'. Otherwise, $H^8(x) \neq H^8(x')$ and $H^8(H^8(x)) = H^8(H^8(x'))$, a collision is found too.

We have:

$$Pr[Succ_{\widetilde{A}}(n)] = Pr[Hash - col_{\widetilde{A},\widetilde{\Pi}}(n) = 1] > \frac{1}{q(n)}$$

$$Pr[Hash - col_{\widetilde{A},\widetilde{\Pi}}(n) = 1|Succ_{\widetilde{A}}(n)] = 1$$

$$
\begin{aligned}
&Pr[Hash - col_{\widetilde{A},\widetilde{\Pi}}(n) = 1 \\
&\quad = Pr[Hash - col_{\widetilde{A},\widetilde{\Pi}}(n) = 1 | Succ_{\widetilde{A}}(n)] * Pr[Succ_{\widetilde{A}}(n)] + Pr[Hash - col_{\widetilde{A},\widetilde{\Pi}}(n) = 1 | \neg Succ_{\widetilde{A}}(n)] * Pr[\neg Succ_{\widetilde{A}}(n)] \\
&\quad \geq Pr[Hash - col_{\widetilde{A},\widetilde{\Pi}}(n) = 1 | Succ_{\widetilde{A}}(n)] * Pr[Succ_{\widetilde{A}}(n)] \\
&\quad = 1 * Pr[Succ_{\widetilde{A}}(n)] \\
&\quad > \frac{1}{q(n)}
\end{aligned} \tag{2}
$$

This completes the reduction as Π is a collision resistant hash function. Therefore our assumption was wrong and $\widetilde{\Pi}$ is collision resistant.

8

9

9.1

Let $k = 0^n$ and $m \in \{0,1\}^n$ be arbitary. Then $f_1(0^n, m) = E(0^n, m)$. Let $k' \in 0, 1^n$ also be arbitary where $k' \neg k$. Then let $m' = E^{-1}(k', E_{0^n}(m) \oplus k') \oplus k'$ So $f_1(0^n, m) = f_1(k', m')$

9.2

9.3

the same as 9.1

10

10.1

The probability that both of them receive the same plate number is $\frac{1}{10} = \frac{1}{2600}$

10.2

$1 \times (1 - \frac{1}{2600}) \times (1 - \frac{2}{2600}) \times ... \times (1 - \frac{n}{2600}) > 1\%$

So $n = 6$, the maximum number of this type of license plates is $n + 1 = 7$.

10.3

$1 \times (1 - \frac{1}{2600 \times 10^m}) \times (1 - \frac{2}{2600 \times 10^m}) \times ... \times (1 - \frac{49}{2600 \times 10^m}) > 1\%$

So $m = 2$, which is the digits should be added at the end of this serial number format.