# Fundamentals of Cryptography Homework 4

周书予
2000013060@stu.pku.edu.cn

October 25, 2022

## Problem 1

### Part A

Obviously $E$ is poly-time computable.

An decoding algorithm counts the number of leading zeros and learns the length of $x$, and finally outputs the last $n$ bits as $x$.

For any $x, x' \in \mathcal{X}^*$,

- if $|x| = |x'|$, because $E(x) \neq E(x')$ and $|E(x)| = |E(x')|$, they are not prefix of each other.
- if $|x| < |x'|$ then $|E(x)| < |E(x')|$, notice that the $|x|+1$ bit in $E(x)$ is 1 while the counterpart is 0 in $E(x')$, thus $E(x)$ is not a prefix of $E(x')$.

### Part B

Write $|x|$ as a binary string of length $\log(|x|)$, then insert a 1 between each adjacent bits, and concatenate with $0\|x$. That is,

$$E(x) = \textsf{insert-one}(x)\|0\|x$$

for example, we have $E(x) = 11010\|0\|x$ for $|x| = 4$ and $E(x) = 1101111\|0\|x$ for $|x| = 11$.

Obviously $E$ is poly-time computable.

An decoding algorithm can learn from the $\textsf{insert-one}$ part (which ends by a 0 on even bits) the length of $x$, and output the last $n$ bits as $x$.

Apparently $E(x) \neq E(x')$ holds for any $x \neq x'$ with the same length. As for $|x| < |x'|$, the $\textsf{insert-one}$ part of encoding makes $E(x)$ and $E(x')$ different in the first $2\log(|x|)$ bits, and thus $E(x)$ is not a prefix of $E(x')$.

### Part D

$$E(x) = \begin{cases} \textsf{int2str}(|x|, n)\|0^{-|x| \bmod n}\|x, & |x| < 2^n - 1 \\ 1^n\|0^{|x|}\|1\|x, & |x| \geqslant 2^n - 1 \end{cases}$$

where function $\textsf{int2str}(m, n)$ converts integer $m$ into a binary string of length $n$.

It's not hard to see that $E$ is a prefix-free encoding, and it suffices $|E(x)| < |x| + 2n$ and $n \mid |E(x)|$ for every $x \in \mathcal{X}^{<2^n-1}$.

## Problem 2

### Part A

Since $F_{CBC}$ is parameterized with a keyed secure PRF $F$, it is not hard to see (we have proven it multiple times) that

$$\left|\Pr\left[\mathcal{D}^{F_{CBC}}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{G_{CBC}}(1^n) = 1\right]\right| < \mathsf{negl}(n)$$

where $G_{CBC}$ is exactly the same as $F_{CBC}$, except a trult random function $g : \{0,1\}^n \to \{0,1\}^n$ is used instead of the PRF $F_k$.

Then we're going to show that

$$\left|\Pr\left[\mathcal{D}^{G_{CBC}}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{f}(1^n) = 1\right]\right| < \mathsf{negl}(n)$$

(where $f$ is a truly random function which maps $(\{0,1\}^n)^*$ to $\{0,1\}^n$) For any prefix-free set $X_1, \cdots, X_q \in (\{0,1\}^n)^*$ queried by the distinguisher and $t_1, \cdots, t_n \in \{0,1\}^n$ the output of the oracle (which is $G_{CBC}$ or $f$), we want to show that

$$\Pr\left[\forall i, G_{CBC}(X_i) = t_i\right] \geqslant (1 - \mathsf{negl}(n)) \Pr\left[\forall i, f(X_i) = t_i\right] \tag{1}$$

If so, for any case where $\mathcal{D}^f$ outputs 1, $\mathcal{D}^{G_{CBC}}$ outputs 1 with probability at least $1-\mathsf{negl}(n)$, and the same when outputing 0. Which means that $\mathcal{D}^{G_{CBC}}$ outputs the same as $\mathcal{D}^f$ with probability at least $1 - \mathsf{negl}(n)$, so they are indistinguishable.

Now we prove eq. (1). For $X_i \in (\{0,1\}^n)^\ell$, we denote $(I_1, I_2, \cdots, I_\ell)$ as $(x_{i,1}, G_{CBC}(x_{i,1}) \oplus x_{i_2}, \cdots, G_{CBC}(x_{i,1}, \cdots, x_{i,\ell-1}) \oplus x_{i,\ell})$, which is all the input to $G_{CBC}$. If there is an $I_i$ in $X$ coincides with another $I'_j$ in $X'$ with different prefix $(x_1, \cdots, x_i) \neq (x'_1, \cdots, x'_j)$, we say it is a **collision**.

It can be proved that

- if there is no collision occurred, then

$$\Pr\left[\forall i, G_{CBC}(X_i) = t_i\right] = \Pr\left[\forall i, f(X_i) = t_i\right]$$

- collision only occurred with negligible probability

These two conclusions are quite intuitive. The first one holds because $g$ is a truly random function, so it outputs identical independent distribution on different inputs. The second one holds because for each pair of $(I_i, I'_j)$, they coincides with negligible probability, which the length and number of $X$ and both polynomial, by union bound we know that even one collision happens with negligible probability.

### Part B

$F_{CBC}$ is a secure prefix-free PRF, which means that

$$|\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi}(n) = 1] - \Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi'}(n) = 1]| < \mathsf{negl}(n)$$

where $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrf})$ is the MAC mentioned in the problem, and $\Pi'$ exactly the same as $\Pi$, except a truly random function $f(\cdot)$ is used instead of $F_{CBC}(k, \cdot)$.

we also have

$$\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi'}(n) = 1] = 2^{-n}$$

which is clear since $E(\cdot)$ outputs different $E(x)$ for different $x$, and any adversary can only make random guessing and has success probability at most $2^{-n}$ in front of the truly random function $f$.

## Problem 3

We define that all operations are over the $\mathbb{F}_{2^n}$ field.

### Part A

- If $i \neq i'$, $m_{j,i} + ik_1 \neq m_{j',i'} + i'k_1$ is equivalent to $k_1 \neq (m_{j,i} - m_{j',i'}) \cdot (i' - i)^{-1}$, which is of probability $1 - \mathsf{negl}(n)$ since $k_1$ is chosen at uniformly random.
- If $i = i'$ then $m_{j,i} \neq m_{j',i'}$, thus $m_{j,i} + ik_1 \neq m_{j',i'} + i'k_1$ must holds.

### Part B

Consider a distinguisher $\mathcal{D}$ which compares $\sum_{i=1}^{\ell_j} \mathcal{O}(m_{j,i} + ik_1)$ with $\sum_{i'=1}^{\ell_{j'}} \mathcal{O}(m_{j',i'} + i'k_1)$, and outputs 1 if equal and 0 otherwise.

For truly random function $f$, it can be seen that $\Pr[\mathcal{D}^{f(\cdot)}(1^n) = 1] = 1 - \mathsf{negl}(n)$ since with probability at least $1 - \mathsf{negl}(n)$, the set $\{m_{j,i} + ik_1\}$ is not equal to $\{m_{j',i'} + i'k_1\}$.

Notice that $F(k_2, \cdot)$ is a PRF, so we also have $\Pr[\mathcal{D}^{F(k_2,\cdot)}(1^n) = 1] = 1 - \mathsf{negl}(n)$, which means that with probability $1 - \mathsf{negl}(n)$,

$$(m_{j,1}, \cdots, m_{j,\ell_j}) \neq (m_{j',1}, \cdots, m_{j',\ell_{j'}}) \Rightarrow \sum_{i=1}^{\ell_j} F(k_2, m_{j,i} + ik_1) \neq \sum_{i'=1}^{\ell_{j'}} F(k_2, m_{j',i'} + i'k_1)$$

### Part C

By replacing $F(k_3, \cdot)$ in $F_{\text{PMAC}}$ with a truly random function $f$, we obtain another function $G_{\text{PMAC}} = g\left(\sum_{i=1}^{\ell} F(k_2, m_i + ik_1)\right)$.

Since $F(k_3, \cdot)$ is a secure PRF, it can be shown that

$$\left|\Pr\left[\mathcal{D}^{F_{\text{PMAX}}}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{G_{\text{PMAC}}}(1^n) = 1\right]\right| < \mathsf{negl}(n)$$

With probability $1 - \mathsf{negl}(n)$, the outputs of $F(k_2, \cdot)$ are all distinct, which means $G_{\text{PMAC}}$ indistinguishable from a truly random function $g$.

Thus, $F_{\text{PMAC}}$ is a secure PRF.

## Problem 4

### Part A

Regard $H(k, (m_1, \cdots, m_\ell))$ as a degree-$\ell$ polynomial of $k$ over $\mathbb{F}_{2^n}$ field.

For distinct messages $m, m'$ with length $\leqslant \ell n$, $H(k, m) - H(k, m')$ is a nonzero polynomial of degree at most $\ell$, which has at most $\ell$ zero points over $\mathbb{F}_{2^n}$.

$\ell$ is polynomial on $n$, which draws the conclusion that

$$\Pr_{k \leftarrow \{0,1\}^n}\left[H(k, m) = H(k, m')\right] \leqslant \frac{\ell}{2^n} = \mathsf{negl}(n)$$

### Part B

Let $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ be the MAC mentioned in the problem, and $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ exactly the same as $\Pi$, except that a truly random function $f$ is used instead of the pseudorandom function $F_{k_2}$. Consider that

$$\Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi'}(n) = 1\right] \leqslant p(n)\mathsf{negl}(n) + (1 - p(n)\mathsf{negl}(n))\, 2^{-n}$$

for some polynomial $p(n)$.

This is because when adversary $\mathcal{A}$ gives $m'$, there is probability at most $p(n)\mathsf{negl}(n)$ such that $H(k, m')$ coincides with some $H(k, m^{(i)})$ asked before, and if there is not, the adversary can only make random guessing and has success probability at most $2^{-n}$.

We can further prove that

$$\left|\Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi}(n) = 1\right] - \Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi'}(n) = 1\right]\right| < \mathsf{negl}(n)$$

Consider a distinguisher $\mathcal{D}$ which emulates the message authentication experiment for $\mathcal{A}$ and observes whether $\mathcal{A}$ succeeds in outputting a valid tag on a “new” message. If so, $\mathcal{D}$ guesses that its oracle is a pseudorandom function; otherwise, it guesses that its oracle is a truly random function.

We have

$$\Pr\left[\mathcal{D}^{F_{k_2}}(1^n) = 1\right] = \Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi}(n) = 1\right]$$
$$\Pr\left[\mathcal{D}^{f}(1^n) = 1\right] = \Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi'}(n) = 1\right]$$

and since $F$ is a secure PRF and $\mathcal{D}$ runs in polynomial time, we also have

$$\left|\Pr\left[\mathcal{D}^{F_{k_2}}(1^n)=1\right]-\Pr\left[\mathcal{D}^{f}(1^n)=1\right]\right|<\mathsf{negl}(n)$$

So finally we draw the conclu that

$$\Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A}.\Pi}(n)=1\right]\leqslant\mathsf{negl}(n)$$

which means that $\Pi$ is a (strongly) secure MAC.

### Part C

If the item $k^\ell$ in $H(k,m)$ is lost, then $H(k,m)=H(k,0^n\|m)$ holds for all $m$, which easily makes this MAC insecure.

## Problem 5

### Part A

Suppose $F'$ is a PRF, then $F(k,x)=\begin{cases}F'(x,0), & k=0^n\\ F'(k,x), & k\neq 0^n\end{cases}$ is also a PRF since it performs differently with $F'$ with only negligible probability.

Thus we have $\hat{F}(k,0^n)=F(k,0^n)\oplus F(0^n,k)=\begin{cases}F'(0^n,0^n)\oplus F'(0^n,0^n) & k=0^n\\ F'(k,0^n)\oplus F'(k,0^n) & k\neq 0^n\end{cases}=0^n$ to be a deterministic value, which makes $\hat{F}$ obviously not a PRF.

### Part B

$\hat{F}(x,y)$ is obviously symmetric, so it is sufficed to prove that $\hat{F}(k,\cdot)$ is a PRF.

For any distinguisher which queries encryption for message $m_1,\cdots,m_{p(n)}$ towards its oracle, there is $1-\mathsf{negl}(n)$ probability that all $g_1(m_i)$-s are distinct since we assume $g_1$ to be collision resistant. And also $F(g_0(k),g_1(m_i))$-s are distinct with $1-\mathsf{negl}(n)$ probability, since $F$ itself is a PRF.

Since $g$ is a PRG, we know that $g_0(k),g_1(m_i)$ should be nearly independent with $g_1(k),g_0(m_i)$, which makes $F(g_0(k),g_1(m_i))$ nearly independent with $F(g_0(m_i),g_1(k))$. In summary, with probability $1-\mathsf{negl}(n)$, all $\hat{F}(k,m_i)$-s are distinct, which gives no information to the distinguisher, and thus it can not distinguish $\hat{F}(k,\cdot)$ from a truly random function.

## Problem 6

### Part A

We prove that $\mathsf{Enc}_1$ is DKMA-secure when $F$ is modeled as an ideal cipher.

For any PPT adversary $\mathcal{A}$, a PPT distinguisher $\mathcal{D}$ can be built, which simulates the interaction between $\mathcal{A}$ and the challenger, except that when $c_i = \mathsf{Enc}(k, x_i)$ is returned, it uses an oracle $\mathcal{O}$ and returns $c_i = (r, \mathcal{O}(r \oplus m))$. Finally, $\mathcal{D}$ outputs 1 if $\mathcal{A}$ wins, and 0 otherwise.

On the one hand, we have the equlity that

$$\Pr\left[\mathsf{DKMA}_{\mathcal{A},\mathsf{Enc}_1}(n) = 1\right] = \Pr\left[\mathcal{D}^{F(k,\cdot)}(1^n) = 1\right]$$

on the other hand, we denote $\mathsf{Enc}_1'$ exactly the same as $\mathsf{Enc}_1$, except a truly random permutation $f(\cdot)$ is in place of $F(k, \cdot)$. When adversary plays DKMA security game under this encryption scheme, the distribution of $\mathsf{Enc}_1'(k, f_{i,0}(k))$ and $\mathsf{Enc}_1'(k, f_{i,1}(k))$ are totally the same, which makes them indistinguishable, so we have

$$\Pr\left[\mathsf{DKMA}_{\mathcal{A},\mathsf{Enc}_1'}(n) = 1\right] = \Pr\left[\mathcal{D}^{f(\cdot)}(1^n) = 1\right] = \frac{1}{2}$$

Since $F(k, \cdot)$ is a ideal cipher, $\mathcal{A}$ can have at most negligible advantage to win this game.

## Part B

For fixed $k_1, k_2$, there is a bijection between permutation $P(m)$ and $Q(m) := F((k_1, k_2), m) = P(m \oplus k_1) \oplus k_2$, which means that the distribution of $P(\cdot)$ and $F((k_1, k_2), \cdot)$ are totally the same, making them indistinguishable.

So $P$ is random permutation implies that $F$ is a strong PRP.

## Part C

We prove that $\mathsf{Enc}_1$ is not DKMA-secure as $F$ defined in **Part B**, which is, $\mathsf{Enc}_1(k, x) = (r, P(r \oplus x \oplus k_1) \oplus k_2)$.

First adversary chooses function $f_{1,0}, f_{1,1}$ such that for all $k = k_1 \| k_2$, $f_{1,0}(k) = f_{1,1}(k) = k_1$, which makes $c_1 = (r, P(r \oplus k_1 \oplus k_1) \oplus k_2) := (c_{1,0}, c_{1,1})$, and adversary can learn $k_2$ by calculating $k_2 = c_{1,1} \oplus P(c_{1,0})$.

Secondly it chooses $f_{2,0}(k) = f_{2,1}(k) = 0^n$, which makes $c_2 = (r, P(r \oplus k_1) \oplus k_2) := (c_{2,0}, c_{2,1})$, and can learn $k_1$ by calculating $k_1 = P^{-1}(c_{2,1} \oplus k_2) \oplus c_{2,0}$.

By now adversary has already learned all information about the key, thus it can easily wins the security game, which means that $\mathsf{Enc}_1$ here is not DKMA-secure.