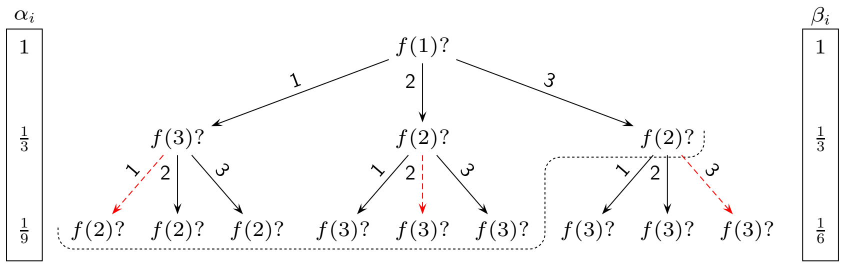
MTAT.07.003 Cryptology II
Spring 2009 / Exercise session IV

PRP/PRF switching lemma



1. Let $\mathcal{A}$ be the adversary that tries to distinguish a random permutation $f : \{1,2,3\} \to \{1,2,3\}$ from a random function $f : \{1,2,3\} \to \{1,2,3\}$ according to the adaptive deterministic querying strategy depicted above. More formally, nodes represents adversaries queries. The adversary $\mathcal{A}$ starts form the root node and moves to next nodes according to the answers depicted as arc labels. The dashed line corresponds to the decision border, where $\mathcal{A}$ stops querying and outputs his or her guess.

   (a) Compute the following probabilities

   $$\begin{aligned}
   &\Pr\left[f \leftarrow \mathcal{F}_{\mathrm{all}} : \mathcal{A} \text{ reaches vertex } u\right] \enspace ,\\
   &\Pr\left[f \leftarrow \mathcal{F}_{\mathrm{all}} : \mathcal{A} \text{ reaches vertex } u \wedge \neg\mathsf{Collision}\right] \enspace ,\\
   &\Pr\left[f \leftarrow \mathcal{F}_{\mathrm{all}} : \neg\mathsf{Collision}\right] \enspace ,\\
   &\Pr\left[f \leftarrow \mathcal{F}_{\mathrm{all}} : \mathcal{A} \text{ reaches vertex } u | \neg\mathsf{Collision}\right] \enspace ,\\
   &\Pr\left[f \leftarrow \mathcal{F}_{\mathrm{prm}} : \mathcal{A} \text{ reaches vertex } u\right]
   \end{aligned}$$

   for all nodes u in the decision border.

   (b) Compute these probabilities for an arbitrary message space $\mathcal{M}$ under the assumption that $\mathcal{A}$ makes exactly q queries and conclude

   $$\Pr\left[\mathcal{A} = 0 | \mathcal{F}_{\mathrm{all}} \wedge \neg\mathsf{Collision}\right] = \Pr\left[\mathcal{A} = 0 | \mathcal{F}_{\mathrm{prm}}\right] \enspace .$$

2. For the proof of the PRP/PRF switching lemma, consider the following games. In the game $\mathcal{G}_0$, the challenger first draws $f \leftarrow \mathcal{F}_{\mathrm{all}}$ and then answers up to q distinct queries. In the game $\mathcal{G}_1$, the challenger draws $f \leftarrow \mathcal{F}_{\mathrm{prm}}$ and then answers up to q distinct queries. In both games, the output is determined by the adversary $\mathcal{A}$ who submits its final verdict.

   (a) Formalise both games as short programs, where $\mathcal{G}$ can make oracle calls to $\mathcal{A}$. For example, something like

$$
\begin{array}{l}
\mathcal{G}_0^{\mathcal{A}} \\
\left[\begin{array}{l}
f \xleftarrow[u]{} \mathcal{F}_{\text{all}} \\
y_0 \leftarrow \perp \\
\text{For } i \in \{1, \ldots, q\} \text{ do} \\
\left[\begin{array}{l}
x_i \leftarrow \mathcal{A}(y_{i-1}) \\
\text{If } x_i = \perp \text{ then break the cycle} \\
y_i \leftarrow f(x_i)
\end{array}\right. \\
\textbf{return } \mathcal{A}
\end{array}\right.
\end{array}
$$

(b) Rewrite both games so that there are no references to the function f but the behaviour does not change. Denote these games by $\mathcal{G}_2, \mathcal{G}_3$.

(c) Analyse what is the probability that execution in the games $\mathcal{G}_2$ and $\mathcal{G}_3$ starts to diverge. Conclude $\mathsf{sd}_\star(\mathcal{G}_2, \mathcal{G}_3) = \Pr[\mathsf{Collision}]$

Hint: Note that following code fragment samples uniformly permutations

$$
\begin{array}{l}
\text{Sample } f(x_i) \\
\left[\begin{array}{l}
y_i \xleftarrow[u]{} \mathcal{M} \\
\text{If } y_i \in \{y_1, \ldots, y_{i-1}\} \text{ then} \\
\left[\, y_i \xleftarrow[u]{} \mathcal{M} \setminus \{y_1, \ldots, y_i\}\right.
\end{array}\right.
\end{array}
$$

What is the probability we ever reach the if branch?

3. Let $y_1, \ldots, y_q$ be chosen uniformly and independently from the set $\mathcal{M}$. Let $\mathsf{Distinct}(k)$ denote the event that $y_1, \ldots, y_k$ are distinct. Estimate the value of $\Pr[\mathsf{Distinct}(k)|\mathsf{Distinct}(k-1)]$ and this result to prove

$$\Pr[\mathsf{Distinct}(k)] \leq e^{-q(q-1)/(2|\mathcal{M}|)}$$

How one can use this result to prove the birthday bound

$$\Pr[\mathsf{Collision}|q \text{ queries}] \geq 0.316 \cdot \frac{q(q-1)}{|\mathcal{M}|} \quad .$$

Hint: Note that $1 - x \leq e^{-x}$.
Hint: Note that $1 - e^{-x} \geq (1 - e^{-1})x$ if $x \in [0, 1]$.

4. A block cipher is commonly modelled as a (t, q, ε)-pseudorandom permutation family $\mathcal{F}$. As such, it is perfect for encrypting a single block.

   (a) The electronic codebook mode ECB uses a same permutation $f \leftarrow \mathcal{F}$ for all message blocks $\mathrm{ECB}_f(m_1\|\ldots\|m_n) = f(m_1)\|\ldots\|f(m_n)$ is known to be insecure pseudorandom permutation. Find an algorithm that can distinguish $\mathrm{ECB}_f : \mathcal{M}^n \rightarrow \mathcal{M}^n$ from a random permutation over $\mathcal{M}^n$. Is this weakness relevant in practise or not?

(b) Let $\mathcal{M}_\circ^n = \{(m_1, \ldots, m_n) \in \mathcal{M}^n : m_i \neq m_j\}$ denote the set of messages with distinct blocks. Show that $\textsc{Ecb}_f : \mathcal{M}_\circ^n \to \mathcal{M}_\circ^n$ is $(t, \frac{q}{n}, \varepsilon)$-pseudorandom permutation family if $\mathcal{F}$ is (t, q, ε)-pseudorandom permutation family.

(c) If addition is defined over $\mathcal{M}$, random shifts $c_1, \ldots, c_n \xleftarrow{u} \mathcal{M}$ can be used to avoid equalities in the message $\overline{\boldsymbol{m}} = (m_1 + c_1, \ldots, m_n + c_n)$. Compute the probability $\Pr[c_1, \ldots, c_n \xleftarrow{u} \mathcal{M} : \overline{\boldsymbol{m}} \notin \mathcal{M}_\circ^n]$.

(d) The cipher-block chaining mode $\textsc{Cbc}$ uses the permutation $f \leftarrow \mathcal{F}$ to link plaintext and ciphertexts: $\textsc{Cbc}_f(m_1\| \ldots \|m_n) = c_1\| \ldots \|c_n$ where $c_i = f(m_i \oplus c_{i-1})$ and c_0 is know as initialisation vector (nonce). The $\textsc{Cbc}$ mode can be viewed as more efficient way to modify the message by setting shifts $c_i \leftarrow f(\overline{m}_{i-1})$. Again, compute the probability $\Pr[c_0 \xleftarrow{u} \mathcal{M}, \cdots, c_n \leftarrow f(m_{n-1} + c_{n-1}) : \overline{\boldsymbol{m}} \notin \mathcal{M}_\circ^n]$. Conclude that CBC_f is a secure pseudorandom permutation over $\mathcal{M}^n$.

5. The IND-CPA security notion is also applicable for symmetric cryptosystems. Namely, a symmetric cryptosystem $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is (t, ε)-IND-CPA secure, if for any t-time adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}(\mathcal{A}) = |\Pr[\mathcal{G}_0^{\mathcal{A}} = 1] - \Pr[\mathcal{Q}_1^{\mathcal{A}} = 1]| \leq \varepsilon$$

where

$$\begin{array}{ll} \mathcal{Q}_0^{\mathcal{A}} & \mathcal{Q}_1^{\mathcal{A}} \\ \left[\begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)} \\ \textbf{return } \mathcal{A}^{\mathcal{O}_1(\cdot)}(\mathsf{Enc}_{\mathsf{sk}}(m_0)) \end{array}\right. & \left[\begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)} \\ \textbf{return } \mathcal{A}^{\mathcal{O}_1(\cdot)}(\mathsf{Enc}_{\mathsf{sk}}(m_1)) \end{array}\right. \end{array}$$

and the oracle $\mathcal{O}_1$ serves encryption calls.

Let $f : \mathcal{M} \times \mathcal{K} \to \mathcal{M}$ be a (t, ε)-pseudorandom permutation. Then a $\textsc{Ctr}$-\$ symmetric encryption scheme is defined as follows:

- A secret key is a randomly chosen $k \xleftarrow{u} \mathcal{K}$.
- To encrypt a message $m_1, \ldots, m_n$, choose a random nonce $s_0 \xleftarrow{u} \mathcal{M}$ and output $s_0, m_1 + f(s_0 + 1, k), \ldots, m_n + f(s_0 + n, k)$.
- To decrypt $s_0, c_1, \ldots, c_n$, output $c_1 - f(s_0 + 1, k), \ldots, c_n - f(s_0 + n, k)$.

Prove that $\textsc{Ctr}$-\$ is IND-CPA secure cryptosystem.

6. Estimate computational distance between following games under the assumption that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is (t, ε)-IND-CPA secure cryptosystem.

(a) Left-or-right games

$$\mathcal{G}_0^{\mathcal{A}} \qquad\qquad\qquad\qquad \mathcal{G}_1^{\mathcal{A}}$$

$$\left[\begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen} \\ \text{ For } i = 1, \ldots, q \text{ do} \\ \left[\begin{array}{l} (m_0^i, m_1^i) \leftarrow \mathcal{A} \\ \text{Give } \mathsf{Enc}_{\mathsf{sk}}(m_0^i) \text{ to } \mathcal{A} \end{array}\right. \\ \textbf{return} \text{ the output of } \mathcal{A} \end{array}\right. \qquad \left[\begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen} \\ \text{ For } i = 1, \ldots, q \text{ do} \\ \left[\begin{array}{l} (m_0^i, m_1^i) \leftarrow \mathcal{A} \\ \text{Give } \mathsf{Enc}_{\mathsf{sk}}(m_1^i) \text{ to } \mathcal{A} \end{array}\right. \\ \textbf{return} \text{ the output of } \mathcal{A} \end{array}\right.$$

(b) Real-or-random games

$$\mathcal{G}_0^{\mathcal{A}} \qquad\qquad\qquad\qquad \mathcal{G}_1^{\mathcal{A}}$$

$$\left[\begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen} \\ \text{ For } i = 1, \ldots, q \text{ do} \\ \left[\begin{array}{l} m^i \leftarrow \mathcal{A} \\ \text{Give } \mathsf{Enc}_{\mathsf{sk}}(m^i) \text{ to } \mathcal{A} \end{array}\right. \\ \textbf{return} \text{ the output of } \mathcal{A} \end{array}\right. \qquad \left[\begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen} \\ \text{ For } i = 1, \ldots, q \text{ do} \\ \left[\begin{array}{l} m_0^i \leftarrow \mathcal{A}, m_1^i \xleftarrow{u} \mathcal{M} \\ \text{Give } \mathsf{Enc}_{\mathsf{sk}}(m_1^i) \text{ to } \mathcal{A} \end{array}\right. \\ \textbf{return} \text{ the output of } \mathcal{A} \end{array}\right.$$