Applied Cryptography

Lecture Note

Spring 2023

Cao, Ganyuan

Contents

1 Security Proof

1.1 Game-based Security Proof Framework

To prove the statment: *"If a scheme F_1 is $\mathcal{S}_1$ secure, then a scheme F_2 is $\mathcal{S}_2$ secure"*, we follow the steps:

1. Suppose by contraposition that there is an adversary $\mathcal{A}$ against $\mathcal{S}_2$ security of F_2 s.t. $\mathbf{Adv}^{\mathcal{S}_2}_{\mathsf{F}_2}(\mathcal{A})$ is not negligible.
2. Construct the adversary $\mathcal{B}$ against $\mathcal{S}_1$ security of F_1 with $\mathcal{A}$ as subroutine.
3. Deduce that $\mathbf{Adv}^{\mathcal{S}_1}_{\mathsf{F}_1}(\mathcal{B})$ is not negligible.

Remarks:

1. Assume that $\mathcal{B}$ is given an oracle $\mathrm{O}_{\mathcal{B}}$, we use $\mathrm{O}_{\mathcal{B}}$ to simulate the pre-defined oracle for $\mathrm{O}_{\mathcal{A}}$. In the adversary $\mathcal{B}$, the adversary $\mathcal{A}$ instead calls the simulation oracle $\mathrm{OSim}_{\mathcal{A}}$.
2. The adversary $\mathcal{B}$ together with the oracle $\mathrm{OSim}_{\mathcal{A}}$ simulates the $\mathcal{S}_2$ security game of F_2.
3. The framework also works for problem reduction. If we want to prove a problem $\mathcal{P}_1$ reduces to a problem $\mathcal{P}_2$, it is equivalent to prove *"if there is an adversary that break the problem $\mathcal{P}_2$ with non-negligible advantage, then there is an adversary $\mathcal{B}$ that break $\mathcal{P}_1$ with non-negligible advantage."*
4. In the case that the primitive $\mathcal{S}_1$ is too "far" from $\mathcal{S}_2$, and *distinguishibility* game in involved, it is better to use "game-chaining" method by decomposing the distinguishibility game into sub-games and chain the sub-games to prove the advantage. Note that the framework proposed by Ballare can be used to write the games for better readability.

1.2 Advantage Rewriting Lemma

Let b be a uniformly random bit, b' be the output of some algorithm. Then

$$\begin{aligned} 2\left|\Pr[b' = b] - \frac{1}{2}\right| &= \left|\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]\right| \\ &= \left|\Pr[b' = 0|b = 0] - \Pr[b' = 0|b = 1]\right| \end{aligned}$$

Proof.

$$\begin{aligned} \Pr[b' = b] - \frac{1}{2} &= \Pr[b' = b \mid b = 1] \cdot \Pr[b = 1] + \Pr[b' = b \mid b = 0] \cdot \Pr[b = 0] - \frac{1}{2} \\ &= \Pr[b' = b \mid b = 1] \cdot \frac{1}{2} + \Pr[b' = b \mid b = 0] \cdot \frac{1}{2} - \frac{1}{2} \\ &= \frac{1}{2}(\Pr[b' = 1 \mid b = 1] + \Pr[b' = 0 \mid b = 0] - 1) \\ &= \frac{1}{2}(\Pr[b' = 1 \mid b = 1] - (1 - \Pr[b' = 0 \mid b = 0])) \\ &= \frac{1}{2}(\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]) \end{aligned}$$

$\square$

1.3 The Difference Lemma

Let Z, W_1, W_2 be (any) events defined over some probability space. Suppose that $\Pr[W_1 \wedge \neg Z] = \Pr[W_2 \wedge \neg Z]$. Then we have $|\Pr[W_2] - \Pr[W_1] \leq \Pr[Z]|$. (In typical uses, we have that $(W_1 \wedge \neg Z)$ occurs if and only if $(W_2 \wedge Z)$ occurs)

Proof.

$$\begin{aligned}
|\Pr[W_2] - \Pr[W_1]| &= |\Pr[(W_1 \wedge Z) \vee (W_1 \wedge \neg Z)] - \Pr[(W_2 \wedge Z) \vee (W_2 \wedge \neg Z)]| \\
&= |\Pr[W_1 \wedge Z] + \Pr[W_1 \wedge \neg Z] - \Pr[W_2 \wedge Z] - \Pr[W_2 \wedge \neg Z]| \\
&= |\Pr[W_1 \wedge Z] - \Pr[W_2 \wedge Z]| \\
&\leq \Pr[Z]
\end{aligned}$$

□

2 Symmetric Encryption

2.1 Symmetric Encryption

A symmetric encryption scheme with key space $\mathcal{K}$, plaintext space $\mathcal{M}$, ciphertext space $\mathcal{C}$, consists of a triple of efficient algoritms: $\mathrm{SE} = (\textsc{KGen}, \textsc{Enc}, \textsc{Dec})$ where

$$\begin{aligned}
&\textsc{KGen} : \{\} \to \mathcal{K}\\
&\textsc{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}\\
&\textsc{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}
\end{aligned}$$

such that

$$\forall K\ \forall m, \textsc{Dec}_K(\textsc{Enc}_K(m)) = m$$

2.2 Block Cipher

A block cipher E with key length k and block size n consists of a pair of efficiently computable permutations $(\mathcal{E}, \mathcal{D})$ where

$$\begin{aligned}
\mathcal{E} &: \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n\\
\mathcal{D} &: \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n
\end{aligned}$$

such that

$$\forall K \in \{0,1\}^k\ \forall m \in \{0,1\}^n, \mathcal{D}(K, \mathcal{E}(K,m)) = m$$

2.3 Pseudorandom Permutation/Function

2.3.1 PRP Security

A block cipher E is defined to be (q,t,ε) secure as a *pseudorandom permutation* (PRP), if for any adversary $\mathcal{A}$ running in time at most t and making at most q queries to $\mathcal{E}_K/\pi$, the advantage $\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) = 2 \cdot |\Pr[\textbf{Game}\ \mathrm{PRP}(\mathcal{A}, E) \Rightarrow \mathsf{true}] - \frac{1}{2}|$$

Game $\mathrm{PRP}(\mathcal{A}, E)$	**Oracle** $\mathrm{RoR}(x)$
1 : $b \leftarrow\$ \{0,1\}$	1 : **if** $b = 0$ **then**
2 : $K \leftarrow\$ \{0,1\}^k$	2 : $\quad y \leftarrow \mathcal{E}_K(x)$
3 : $\pi \leftarrow\$ \mathsf{Perms}[\{0,1\}^n]$	3 : **else**
4 : $b' \leftarrow\$ \mathcal{A}^{\mathrm{RoR}}()$	4 : $\quad y \leftarrow \pi(x)$
5 : **return** $b' = b$	5 : **return** y

Figure 1: PRP Game

Remark:

1. Subsaction of $\frac{1}{2}$ to measure how much better than random guessing the adversary $\mathcal{A}$ does.

2. Scaling factor 2 turns the advantage into a number in the range $[0, 1]$.

3. Here the oracle RoR refers to "real or random". In the real world ($b = 0$), the block cipher E is used. In ideal world ($b = 1$), a random permutation π is used.

2.3.2 PRF Security

A block cipher E is defined to be (q, t, ε)-secure as a *pseudorandom function* (PRF), if for any adversary $\mathcal{A}$ running in time at most t and making at most q queries to $\mathcal{E}_K/\rho$, the advantage $\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) = 2 \cdot |\Pr[\mathbf{Game}\ \mathrm{PRF}(\mathcal{A}, E) \Rightarrow \mathsf{true}] - \frac{1}{2}|$$

$\mathbf{Game}$ $\mathrm{PRF}(\mathcal{A}, E)$	$\mathbf{Oracle}$ $\mathrm{RoR}(x)$
$b \leftarrow\!\$\ \{0,1\}$	$\mathbf{if}\ b = 0\ \mathbf{then}$
$K \leftarrow\!\$\ \{0,1\}^k$	$\quad y \leftarrow \mathcal{E}_K(x)$
$\rho \leftarrow\!\$\ \mathsf{Funcs}[\{0,1\}^n]$	$\mathbf{else}$
$b' \leftarrow\!\$\ \mathcal{A}^{\mathrm{RoR}}()$	$\quad y \leftarrow \rho(x)$
$\mathbf{return}\ b' = b$	$\mathbf{return}\ y$

Figure 2: PRF Game

Remarks:

1. A pseudorandom Function (PRF) is a function $\rho : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ defined over $(\mathcal{K}, \mathcal{D}, \mathcal{R})$ s.t. $\rho(k, x)$ can be evaluated efficiently.

2. A pseudorandom Permutation (PRP) is a permutation $\pi : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ defined over $(\mathcal{K}, \mathcal{D}, \mathcal{R})$ s.t. $\pi(k, x)$ can be evaluated efficiently; $\pi(k, \cdot)$ is injective; there exists an efficient inversion algorithm π^{-1}.

3. The difference between a PRF and a PRP is that PRF does *lazy sampling.* So a PRF may sample $y_i, y_j \in \mathcal{R}$ such that $y_i = y_j$ for some $i \neq j$. Suppose q queries are made to a PRF, there are $\frac{q(q-1)}{2}$ such pairs and each happens with probability $\frac{1}{|\mathcal{R}|}$. Thus such event happens with probability $\frac{q(q-1)}{2|\mathcal{R}|}$.

2.3.3 PRP-PRF Switching Lemma

Let E be a bock cipher. Then for any adversary $\mathcal{A}$ making q queries,

$$|\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) - \mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A})| \leq \frac{q^2}{2^{n+1}}$$

Proof. Let $\mathcal{A}$ be an (q, t, ε) adversary that plays the game $\mathrm{G}_0 - \mathrm{G}_2$ in Figure 3. We have that $\mathrm{G}_0 = \mathrm{G}_E^{\mathrm{PRP}} = \mathrm{G}_E^{\mathrm{PRF}}$, $\mathrm{G}_1 = \mathrm{G}_\pi^{\mathrm{PRP}}$, $\mathrm{G}_2 = \mathrm{G}_\rho^{\mathrm{PRF}}$. Thus we have that

$$\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) = \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_1(\mathcal{A})]$$

and

$$\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) = \Pr[\mathrm{G}_0(\mathcal{A})] - \Pr[\mathrm{G}_2(\mathcal{A})]$$

Hence,

$$\left|\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) - \mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A})\right| \leq \left|\Pr[\mathrm{G}_2(\mathcal{A})] - \Pr[\mathrm{G}_1(\mathcal{A})]\right|$$

We have that G_1 and G_2 are identical unless a repeated value occurs amongst the output values in G_2. Consider that in game G_2, the adversary queries q times. Thus we need to sample q output values y_i uniformly at random from $\{0,1\}^n$. Thus $\Pr[y_i = y_j] = 2^{-n}$ for each pair of (i, j). There are $\frac{q(q-1)}{2} \leq \frac{q^2}{2}$ pairs of indices. By union bound, we have that

$$\Pr[y_i = y_j \text{ for some } i \neq j] \leq \frac{q^2}{2} \cdot 2^{-n} = \frac{q^2}{2^{n+1}}$$

By the Difference Lemma, we have that

$$\begin{aligned}\left|\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) - \mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A})\right| &\leq \left|\Pr[\mathrm{G}_2(\mathcal{A})] - \Pr[\mathrm{G}_1(\mathcal{A})]\right| \\ &\leq \frac{q^2}{2^{n+1}}\end{aligned}$$

□

Game G_0	**Game** G_1	**Game** G_1
procedure INIT	**procedure** INIT	**procedure** INIT
$K \leftarrow\!\!\$\ \mathcal{K}$	$\pi \leftarrow\!\!\$\ \mathcal{P}[\{0,1\}^n]$	$\rho \leftarrow\!\!\$\ \mathcal{F}[\{0,1\}^n]$
procedure OEnc(m)	**procedure** OEnc(m)	**procedure** OEnc(m)
return $E(K, m)$	**return** $\pi(m)$	**return** $\rho(m)$

Figure 3: Proof of PRP/PRF Switching Lemma

Remark: This leads to the following game that on an adversary's distinguishibility between a pseudorandom permutation and a pseudorandom function. The advantage is

$$\mathbf{Adv}^{\mathrm{PRP/PRF}}(\mathcal{A}) = \frac{q^2}{2^{n+1}}$$

Game PRP/PRF($\mathcal{A}$)

$b \leftarrow\!\!\$ \{0,1\}$
$K \leftarrow\!\!\$ \mathcal{K}$
$\Pi \leftarrow\!\!\$ \mathsf{Perms}[\{0,1\}^n]$
$F \leftarrow\!\!\$ \mathsf{Funcs}[\{0,1\}^n]$
$b' \leftarrow\!\!\$ \mathcal{A}^{\mathrm{LoR}}()$
return $b' = b$

Oracle LoR(x)

if $b = 0$ **then**
  $y \leftarrow \Pi(x)$
elseif $b = 1$ **then**
  $y \leftarrow F(x)$
return y

Figure 4: PRP / PRF Game

2.4 Ciphertext/Plaintext Integrity

2.4.1 INT-CTXT Security

A symmetric encryption scheme SE is said to be (q_e, t, ε)-*ciphertext integrity* (INT-CTXT) secure, if for any adversary $\mathcal{A}$ running in time t and making at most q_e encryption oracle queries and exact one try query to oracle OTry, the advantage $\mathbf{Adv}_{\mathrm{SE}}^{\mathrm{INT-CTXT}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}_{\mathrm{SE}}^{\mathrm{INT-CTXT}}(\mathcal{A}) = \Pr[\textbf{Game}\ \text{INT-CTXT} \Rightarrow 1]$$

Game INT-CTXT($\mathcal{A}$, SE)

$K \leftarrow\!\!\$ \textsc{KGen}(1^\lambda)$
$\mathcal{Q} \leftarrow \emptyset$
$\mathcal{A}^{\mathrm{OEnc,OTry}}()$
return win

Oracle OEnc(m)

$c \leftarrow \textsc{Enc}(K, m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{c\}$
return c

Oracle OTry(c^*)

$\mathsf{win} \leftarrow 0$
$m^* \leftarrow \textsc{Dec}(K, c^*)$
if $c^* \notin \mathcal{Q} \wedge m^* \neq \perp$ **then**
  $\mathsf{win} \leftarrow 1$

Figure 5: INT-CTXT Game

2.4.2 INT-PTXT Security

A symmetric encryption scheme SE is said to be (q_e, t, ε)-*plaintext integrity* (INT-PTXT) secure if for all adversary $\mathcal{A}$ running in time t and making at most q_e encryption oracle queries with $\mathbf{Adv}_{\mathrm{SE}}^{\mathrm{INT-PTXT}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}_{\mathsf{SE}}^{\mathrm{INT-PTXT}}(\mathcal{A}) = \Pr[\textbf{Game}\ \text{INT-PTXT} \Rightarrow 1]$$

Game INT-PTXT($\mathcal{A}$, SE)

$K \leftarrow\$ \textsc{KGen}(1^\lambda)$
$\mathcal{Q} \leftarrow \emptyset$
$\mathcal{A}^{\mathrm{OEnc},\mathrm{OTry}}()$
return win

Oracle OEnc(m)

$c \leftarrow \textsc{Enc}(K, m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
return c

Oracle OTry(c^*)

win $\leftarrow 0$
$m^* \leftarrow \textsc{Dec}(K, c^*)$
if $m^* \notin \mathcal{Q} \wedge m^* \neq \perp$ **then**
  win $\leftarrow 1$

Figure 6: INT-PTXT Game

2.4.3 INT-CTXT > INT-PTXT

If a symmetric encryption scheme SE is INT-CTXT secure, then it is also INT-PTXT secure.

Proof. We prove by contraposition that if SE is not INT-CTXT, then it is not INT-PTXT. Let $\mathcal{A}$ be a INT-CTXT adversary against SE, we construct an adversary $\mathcal{B}$ against INT-PTXT of SE such that $\mathcal{B}$ runs $\mathcal{A}$ and replys $\mathcal{A}$'s queries to $\mathcal{B}$'s OEnc and OTry.

We have that $\mathcal{B}$ simulates the INT-CTXT game of $\mathcal{A}$ since $\mathcal{B}$ makes the exact the same number of queries as $\mathcal{A}$ and $\mathcal{B}$ returns the same c^* as $\mathcal{A}$.

We have that $\mathcal{B}$ wins if $\mathcal{A}$ wins. Let c^* be the ciphertext query $\mathcal{A}$ makes to OTry. Since $\mathcal{A}$ wins, we have that $c^* \notin \mathcal{Q}_c$, which implies $m^* \notin \mathcal{Q}_m$ where $m^* = \textsc{Dec}(K, c^*)$. Thus $\mathcal{B}$ wins if $\mathcal{A}$ wins.

□

2.5 Ciphertext Indistinguishability

2.5.1 IND-CPA Security

A symmetric encryption scheme SE is defined to be (q, t, ε)-*indistinguishability under chosen plaintext attack* (IND-CPA) secure, if for any adversaries $\mathcal{A}$ running in time at most t and making at most q encryption queries, the advantage $\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}}(\mathcal{A}) = 2 \cdot |\Pr[\textbf{Game } \text{IND-CPA}(\mathcal{A}, \text{SE}) \Rightarrow \mathsf{true}] - \frac{1}{2}|$$

Game IND-CPA($\mathcal{A}, \mathrm{SE}$)	**Oracle** LoR(m_0, m_1)
$b \leftarrow\$ \{0,1\}$	**if** $\lvert m_0 \rvert \neq \lvert m_1 \rvert$ **then**
$K \leftarrow\$ \textsc{KGen}(1^\lambda)$	**return** $\perp$
$b' \leftarrow\$ \mathcal{A}^{\mathrm{LoR}}()$	$c \leftarrow\$ \textsc{Enc}_K(m_b)$
return $b' = b$	**return** c

Figure 7: IND-CPA Game

Remarks:

1. IND-CPA security imples decryption security.
2. IND-CPA security implies key recovery (TKR) security.
3. IND-CPA security ensures that every bit of the plaintext is hidden.
4. One-time Pad is IND-CPA is 1-query IND-CPA secure.
5. Here oracle LoR refers to "left or right".
6. A special form of IND-CPA security, which formalize the indistinguishability of a symmetric encryption scheme from random bits, named IND\$-CPA, is defined as in Figure 8.

Game IND\$-CPA($\mathcal{A}, \mathrm{SE}$)	**Oracle** RoR(m)
$b \leftarrow\$ \{0,1\}$	**if** $b = 0$ **then**
$K \leftarrow\$ \textsc{KGen}(1^\lambda)$	$c \leftarrow\$ \textsc{Enc}(K, m)$
$b' \leftarrow\$ \mathcal{A}^{\mathrm{RoR}}()$	**else**
return $b' = b$	$c \leftarrow\$ \mathcal{C}$
	return c

Figure 8: IND\$-CPA Game

2.5.2 IND-CCA Security

A symmetric encryption scheme SE is defined to be $(q_e, q_d, t, \varepsilon)$-*indistinguishibility under chosen ciphertext attack* secure (IND-CCA), if for any adversaries $\mathcal{A}$ running in time at most t and making at most q_e encryption queries to oracle LoR and at most q_d decryption queries to oracle ODec, the advantage $\mathbf{Adv}_{\mathrm{SE}}^{\text{IND-CPA}}(\mathcal{A}) \leq \varepsilon$.

$$\mathbf{Adv}_{\mathrm{SE}}^{\text{IND-CCA}}(\mathcal{A}) = 2 \cdot |\Pr[\mathbf{Game}\ \text{IND-CCA}(\mathcal{A}, \mathrm{SE}) \Rightarrow \mathsf{true}] - \frac{1}{2}|$$

Game IND-CCA($\mathcal{A}$, SE)	**Oracle** LoR(m_0, m_1)	**Oracle** ODec(c)
$b \leftarrow\$ \{0,1\}$	**if** $|m_0| \neq |m_1|$ **then**	**if** $c \in \mathcal{Q}$ **then**
$K \leftarrow\$ \text{KGen}(1^\lambda)$	**return** $\perp$	**return** $\perp$
$\mathcal{Q} \leftarrow \emptyset$	$c \leftarrow\$ \text{Enc}_K(m_b)$	$m \leftarrow \text{Dec}(K, c)$
$b' \leftarrow\$ \mathcal{A}^{\text{LoR,ODec}}()$	$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{c\}$	**return** m
return $b' = b$	**return** c	

Figure 9: IND-CCA Game

2.6 Authenticated Encryption

2.6.1 AE Security

A symmetric encryption scheme SE is said to be *authenticated encryption* (AE) if it is IND-CPA secure and an adversary $\mathcal{A}$ with access to an encryption oracle cannot forge any new ciphertexts i.e.,

$$\text{AE} := \text{IND-CPA} + \text{INT-CTXT}$$

2.6.2 Nonce-based AEAD

A nonce-based AEAD scheme with key space $\mathcal{K}$, message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, nonce space $\mathcal{N}$, and associated data space $\mathcal{AD}$, consists of a triple of algorithms (KGen, Enc, Dec) where:

$$\begin{aligned} &\text{KGen} : \{\} \to \mathcal{K} \\ &\text{Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{M} \to \mathcal{C} \\ &\text{Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\} \end{aligned}$$

such that:

$$\forall k \in \mathcal{K}\ \forall m \in \mathcal{M}\ \forall N \in \mathcal{N}\ \forall AD \in \mathcal{AD}, \text{Dec}(K, N, AD, \text{Enc}(K, N, AD, m)) = m$$

2.7 Case Study: Prove CTR mode is IND-CPA

In this section, we prove that the following theorem:

Theorem 1. *Let $\mathcal{A}$ be an* IND-CPA *adversary against the (simplified) CTR mode* SE *based on a block cipher E, then we can construct a* PRP *adversary $\mathcal{B}$ against E such that*

$$\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}_{\text{CTR}}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\text{PRP}}_{E}(\mathcal{B}) + \frac{q^2}{2^{n-1}}$$

Proof. Consider the games $\text{G}_0 - \text{G}_3$ defined in Figure 10. Let W_i be the event that $b = b'$ in G_i respectively, we have that

$$\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}_{\text{CTR}}}(\mathcal{A}) = \mathbf{Adv}^{\text{G}_0}_{\text{SE}_{\text{CTR}}}(\mathcal{A}) = 2 \cdot \left| \Pr[W_0] - \frac{1}{2} \right|$$

Note that we have

$$\begin{aligned}\left|\Pr[W_0] - \frac{1}{2}\right| &= \left|(\Pr[W_0] - \Pr[W_1]) + (\Pr[W_1] - \Pr[W_2]) + (\Pr[W_2] - \Pr[W_3]) + (\Pr[W_3] - \frac{1}{2})\right| \\ &\leq |(\Pr[W_0] - \Pr[W_1])| + |(\Pr[W_1] - \Pr[W_2])| + |(\Pr[W_2] - \Pr[W_3])| + \left|(\Pr[W_3] - \frac{1}{2})\right|\end{aligned}$$

Since in G_3, we have that the encryption is done via a OTP, which has perfect secrecy, we have that

$$\mathbf{Adv}^{\mathrm{G}_3}_{\mathrm{SE}_{\mathrm{CTR}}}(\mathcal{A}) = 2 \cdot \left|\Pr[W_3] - \frac{1}{2}\right| = 0$$

Thus we have that

$$\left|\Pr[W_0] - \frac{1}{2}\right| \leq |(\Pr[W_0] - \Pr[W_1])| + |(\Pr[W_1] - \Pr[W_2])| + |(\Pr[W_2] - \Pr[W_3])|$$

We first want to show that $|\Pr[W_0] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathrm{PRP}}_{E}(\mathcal{B})$ for some PRP adversary $\mathcal{B}$ against the block cipher E. We define $\mathcal{B}$ as in Figure 11. Observe that $\mathcal{B}$ makes the same number of queries as $\mathcal{A}$ makes, $\mathcal{B}$ internally flips a coin and uses its own RoR oracle to simulate the queries $\mathcal{A}$ makes to the LoR oracle. Also, the running time of $\mathcal{B}$ is essentially of $\mathcal{A}$. Thus we have that $\mathcal{B}$ perfectly simuate the IND-CPA that $\mathcal{A}$ plays. Let d be the secret bit in the PRP game that $\mathcal{B}$ plays, we have that

$$\Pr[W_0] = \Pr[b' = b \mid \mathrm{G}_0(\mathcal{A})] = \Pr[b = b' \mid d = 0] = \Pr[d' = 0 \mid d = 0]$$

and

$$\Pr[W_1] = \Pr[b' = b \mid \mathrm{G}_1(\mathcal{A})] = \Pr[b = b' \mid d = 0] = \Pr[d' = 0 \mid d = 1]$$

By Advantage Rewriting Lemma, we have that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{PRP}}_{E}(\mathcal{B}) &= \left|\Pr[d' = 0 | d = 0] - \Pr[d' = 0 | d = 1]\right| \\ &= |\Pr[W_0] - \Pr[W_1]|\end{aligned}$$

We then prove that $|\Pr[W_1] - \Pr[W_2]| \leq \frac{q^2}{2^{n+1}}$ where q denotes the number of queries. We construct an adversary $\mathcal{B}_1$ that distinguish between a random permutation and a random function as in Figure 12. Note that $\mathcal{B}_1$ makes the same number of queries that $\mathcal{A}$ does, $\mathcal{B}_1$ flips the coin internally and simulates the LoR oracle queries made by $\mathcal{A}$ with its own oracle RoR. Also, $\mathcal{B}_1$ runs in the essentially the same time as $\mathcal{A}$. Thus $\mathcal{B}_1$ perfectly simulates the IND-CPA game that $\mathcal{A}$ plays. Let d be the secret bit in the $\mathrm{PRP} - \mathrm{PRF}$ game that $\mathcal{B}_1$ plays, we have that:

$$\Pr[W_1] = \Pr[b' = b \mid \mathrm{G}_1(\mathcal{A})] = \Pr[b = b' \mid d = 0] = \Pr[d' = 0 \mid d = 0]$$

and

$$\Pr[W_2] = \Pr[b' = b \mid \mathrm{G}_2(\mathcal{A})] = \Pr[b = b' \mid d = 0] = \Pr[d' = 0 \mid d = 1]$$

By Advantage Rewriting Lemma and PRP-PRF Switching Lemma, we have that:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{PRP}}_{E}(\mathcal{B}) &= \left|\Pr[d' = 0 | d = 0] - \Pr[d' = 0 | d = 1]\right| \\ &= |\Pr[W_1] - \Pr[W_2]| \\ &\leq \frac{q^2}{2^{n+1}}\end{aligned}$$

We finally want to show that $\Pr[W_2] - \Pr[W_3] \leq \frac{q^2}{2^{n+1}}$. We construct a IND-CPA challenger $\mathcal{B}_2$. Define $\mathcal{B}_2$ as in Figure 13. Observe that G_2 and G_3 are identical unless the randomly chosen values for *ctr* are not all distinct. Let Z be such event. We have that $(W_2 \wedge \neg Z)$ happens if and only if $(W_3 \wedge \neg Z)$ occurs. Similarly, we have that $\Pr[Z] = \frac{q(q-2)}{2^{n+1}} \leq \frac{q^2}{2^{n+1}}$. Thus by Difference Lemma, we have that

$$|\Pr[W_2] - \Pr[W_3]| \leq \Pr[Z] \leq \frac{q^2}{2^n + 1}$$

Finally, we have that:

$$\begin{aligned}
\mathbf{Adv}^{\text{IND-CPA}}_{\text{SE}_{\text{CTR}}}(\mathcal{A}) &= 2 \cdot \left| \Pr[W_0] - \frac{1}{2} \right| \\
&\leq 2 \cdot |(\Pr[W_0] - \Pr[W_1])| + 2 \cdot |(\Pr[W_1] - \Pr[W_2])| + 2 \cdot |(\Pr[W_2] - \Pr[W_3])| \\
&\leq 2 \cdot \mathbf{Adv}^{\text{PRP}}_{E}(\mathcal{B}) + \frac{2q^2}{2^{n+1}} + \frac{2q^2}{2^{n+1}} \\
&= 2 \cdot \mathbf{Adv}^{\text{PRP}}_{E}(\mathcal{B}) + \frac{q^2}{2^{n-1}}
\end{aligned}$$

□

Game G_0 $\boxed{G_1}$ G_2 $\boxed{G_3}(\mathcal{A}, \text{SE})$

$b \leftarrow\$ \{0,1\}$
$K \leftarrow\$ \text{KGen}(1^\lambda)$
$\boxed{\pi \leftarrow\$ \mathcal{P}(\{0,1\}^n)}$
$\rho \leftarrow\$ \mathcal{F}(\{0,1\}^n)$
$b' \leftarrow\$ \mathcal{A}^{\text{LoR}_0}()$
return $b' = b$

Oracle $\text{LoR}(m_0, m_1)$

$ctr \leftarrow\$ \{0,1\}^n$
$r \leftarrow E_K(ctr)$
$\boxed{r \leftarrow \pi(ctr)}$
$r \leftarrow \rho(ctr)$
$\boxed{r \leftarrow\$ \{0,1\}^n}$
$c_0 \leftarrow m_b \oplus r$
$c \leftarrow ctr || c_0$
return c

Figure 10: Game for IND-CPA CTR proof

Adversary $\mathcal{B}^{\text{RoR}}$

$b \leftarrow\$ \{0,1\}$
$b' \leftarrow\$ \mathcal{A}^{\text{LoR}_{\text{SIM}}}()$
if $b = b'$ **then**
  return 0
else
  return 1

Oracle $\text{RoR}(m)$

if $b = 0$ **then**
  return $E_K(m)$
else
  return $\pi(m)$

Oracle $\text{LoR}_{\text{SIM}}(m_0, m_1)$

$ctr \leftarrow\$ \{0,1\}^n$
$r \leftarrow \text{RoR}(ctr)$
$c_0 \leftarrow m_b \oplus r$
$c \leftarrow ctr || c_0$
return c

Figure 11: Adversary $\mathcal{B}$ for IND-CPA CTR proof

Adversary $\mathcal{B}_1^{\text{RoR}}$

$b \leftarrow\$ \{0,1\}$
$b' \leftarrow\$ \mathcal{A}^{\text{LoR}_{\text{SIM}}}()$
if $b = b'$ **then**
    return 0
else
    return 1

Oracle $\text{RoR}(m)$

if $b = 0$ **then**
    return $\pi(m)$
else
    return $\rho(m)$

Oracle $\text{LoR}_{\text{SIM}}(m_0, m_1)$

$ctr \leftarrow\$ \{0,1\}^n$
$r \leftarrow \text{RoR}(ctr)$
$c_0 \leftarrow m_b \oplus r$
$c \leftarrow ctr || c_0$
return c

Figure 12: Adversary $\mathcal{B}_1$ for IND-CPA CTR proof

Challenger $\mathcal{B}_2$

$b \leftarrow\$ \{0,1\}$

Oracle $\text{RoR}(m)$

if $b = 0$ **then**
    return $\rho(m)$
else
    $r \leftarrow\$ \{0,1\}^n$
    return r

Oracle $\text{LoR}_{\text{SIM}}(m_0, m_1)$

$ctr \leftarrow\$ \{0,1\}^n$
$r \leftarrow \text{RoR}(ctr)$
$c_0 \leftarrow m_b \oplus r$
$c \leftarrow ctr || c_0$
return c

Figure 13: Challenger $\mathcal{B}_2$ for IND-CPA CTR proof

2.8 Case Study: CBC Padding Oracle Attack

The CBC mode of encryption is defined as:

$\mathsf{CBC}[E].\mathcal{E}_K(M_1 || \cdots || M_\ell)$

$C_0 \leftarrow\$ \{0,1\}^n$
for $i = 1, \cdots, \ell$ **do**
    $C_i \leftarrow E_K(M_i \oplus C_{i-1})$
return $C_0 || C_1 || \cdots || C_\ell$

$\mathsf{CBC}[E].\mathcal{D}_K(C_0 || C_1 || \cdots || C_\ell)$

for $i = 1, \cdots, \ell$ **do**
    $M_i \leftarrow E_K^{-1}(C_i) \oplus C_{i-1}$
return $C_0 || C_1 || \cdots || C_\ell$

First is to recover the *Last Byte*. Let pad denote the minimum possible padding byte of a legitimate padding scheme, to recovery $M_\ell[n]$, follow the process

$\mathsf{Padding\text{-}Oracle\text{-}Last\text{-}Byte}$

for $i = \texttt{0x00}, \cdots, \texttt{0xff}$ **do**
    $C'_{\ell-1} \leftarrow C_{\ell-1} \oplus (\texttt{0x00}||...||i)$
    $C' \leftarrow C_0||...||C_{\ell-1}||C_\ell$
    $\mathsf{good\text{-}pad} \leftarrow \text{Padding}(C')$
    if $\mathsf{good\text{-}pad} = \mathsf{true}$ **then**
        $v \leftarrow (\texttt{0x00}||...||i) \oplus (\texttt{0x00}||...||\mathsf{pad})$
        return $v[n]$

Denote $\Delta_{\ell,n}$ as the value of i such that $\textsf{good-pad}$ is set to $\textsf{true}$, according to the decryption scheme, we have that

$$\begin{aligned}
C_{\ell-1}[n] \oplus \Delta_{\ell,n} \oplus E^{-1}(C_\ell)[n] &= \textsf{pad} \\
C_{\ell-1}[n] \oplus E^{-1}(C_\ell)[n] &= \textsf{pad} \oplus \Delta_{\ell,n} \\
M_\ell[n] &= \textsf{pad} \oplus \Delta_{\ell,n}
\end{aligned}$$

Then we can recover the full block following the similar strategy. Let $\textsf{pad}' = \textsf{pad} + 1$, compute $\Delta'_{\ell,n} = \Delta_{\ell,n} \oplus \textsf{pad}'$ and $C'_\ell = C_\ell \oplus (\texttt{0x00}||\cdots||\Delta'_{\ell,n})$ and run the above process again. Note this time, we have

$$\begin{aligned}
(C_{\ell-1}[n-1]||C_{\ell-1}[n]) \oplus (\Delta_{\ell,n-1}||\Delta'_{\ell,n}) \oplus (E^{-1}(C_\ell)[n-1]||E^{-1}(C_\ell)[n]) &= \textsf{pad}'||\textsf{pad}' \\
(C_{\ell-1}[n-1]||C_{\ell-1}[n]) \oplus (E^{-1}(C_\ell)[n-1]||E^{-1})(C_\ell)[n]) &= \textsf{pad}'||\textsf{pad}' \oplus \Delta_{\ell,n-1}||\Delta'_{\ell,n} \\
M_\ell[n-1] &= \textsf{pad} \oplus \Delta_{\ell,n-1}
\end{aligned}$$

3 Hash Function

3.1 Hash Function

A *(cryptographic) hash function* H with message space $\mathcal{M}$ and digest space $\mathcal{T}$ is an efficiently computable function $H : \mathcal{M} \to \mathcal{T}$ mapping an arbitrary length input string to a fixed-length message digest. A *keyed hash function* H with key space $\mathcal{K}$, message space $\mathcal{M}$ and digest space $\mathcal{T}$ is deterministic algorithm that takes two inputs, a key $K \in \mathcal{K}$ and a message $m \in \mathcal{M}$ and output $t := H(K, m) \in \mathcal{T}$.

3.2 Security Goals for Hash Function

3.2.1 Informal Definition

- *Primary Security Goals*

  1. Pre-image Resistance (one-wayness): Given h, it is infeasible to find $m \in \{0,1\}^*$ such that $H(m) = h$. (See *Digital Signature Lecture Note* for adversary-based definition).
  2. Second Pre-image Resistance: given m_1, it is infeasible to find $m_2 \neq m_1$ such that $H(m_1) = H(m_2)$.
  3. Collision Resistance: it is infeasible to find $_1 \neq m_2$ such that $H(m_1) \neq H(m_2)$.

- *Secondary Security Goals*

  1. Near-collision Resistance: it is infeasible to find $m_1 \neq m_2$ such that $H(m_1) \approx H(m_2)$.
  2. Partial Pre-image Resistance 1: given $H(m)$, it is infeasible to recover any partial information about m.
  3. Partial Pre-image Resistance 2: given a target string t of bit-length ℓ, it is infeasible to find $m \in \{0,1\}^*$ such that $H(m) = t||z$ in time significantly faster than 2^ℓ hash evluations.

3.2.2 Collision Resistance

Let $H : \mathcal{D} \to \mathcal{R}$ be a hash function. An algorithm $\mathcal{A}$ is said to be (t, ε) *collision resistance* (CR) adversary against H if $\mathcal{A}$ runs in time t with advantage

$$\mathbf{Adv}_H^{\mathrm{CR}}(\mathcal{A}) = \Pr[\mathbf{Game}\ \mathrm{CR} \Rightarrow 1] = \varepsilon$$

Game CR$(\mathcal{A}, H)$

$(m, m') \leftarrow\!\!\$\; \mathcal{A}()$
if $m \neq m' \wedge H(m) = H(m')$ **then**
  return 1
else
  return 0

Figure 14: Collision Resistance (CR) Game

Remarks:

1. Collision must exist because $|\mathcal{D}| \gg |\mathcal{R}|$.
2. Fix a hash function H, there must be an efficient algorithm $\mathcal{A}$ that outputs collisions.
3. Thus we cannot have a security definition for collision resistance that quantifies over all efficient algorithms $\mathcal{A}$.

3.2.3 Second Pre-image Resistance

Let $H : \mathcal{D} \to \mathcal{R}$ be a hash function. An algorithm $\mathcal{A}$ is said to be (t, ε) *second pre-image resistance* (2PRE) adversary against H if $\mathcal{A}$ runs in time t with advantage

$$\mathbf{Adv}_H^{\mathrm{2PRE}}(\mathcal{A}) = \Pr[\mathbf{Game}\ \mathrm{2Pre} \Rightarrow 1] = \varepsilon$$

Game 2Pre$(\mathcal{A}, H)$

$m \leftarrow\!\!\$\; \mathcal{D}$
$h \leftarrow H(m)$
$m' \leftarrow\!\!\$\; \mathcal{A}(m, h)$
if $m \neq m' \wedge H(m') = h$ **then**
  return 1
else
  return 0

Figure 15: Second Preimage Resistance (2Pre) Game

3.2.4 Pre-image Resistance

Let $H : \mathcal{D} \to \mathcal{R}$ be a hash function. An algorithm $\mathcal{A}$ is said to be (t, ε) *pre-image resistance* ((r)PRE) adversary against H if $\mathcal{A}$ runs in time t with advantage

$$\mathbf{Adv}_H^{\mathrm{(R)PRE}}(\mathcal{A}) = \Pr[\mathbf{Game}\ \mathrm{(r)Pre} \Rightarrow 1] = \varepsilon$$

Game rPre$(\mathcal{A}, H)$	**Game** Pre$(\mathcal{A}, H)$
$h \leftarrow\$ \mathcal{R}$	$m \leftarrow\$ \mathcal{D}$
$m \leftarrow\$ \mathcal{A}(h)$	$h \leftarrow H(m)$
if $H(m) = h$ **then**	$m' \leftarrow\$ \mathcal{A}(m, h)$
return 1	**if** $H(m') = h$ **then**
else	**return** 1
return 0	**else**
	return 0

Figure 16: rPre and Pre Game

Remarks:

1. The notation PRE is also denoted as *one-wayness*. We then say that H is a *one-way function* (OWF).

3.2.5 CR $>$ 2Pre

Any hash function that is collision resistant is also second pre-image-resistant

Proof. Assume by contraposition a hash function H is not second pre-image-resistance, we want to prove that H is not collision resistant. Let $\mathcal{A}$ be an adversary against second pre-image resistance of H, we want to construct an $\mathcal{B}$ against collision resistance of H. Define $\mathcal{B}$ as follows:

Adversary $\mathcal{B}$
$m \leftarrow\$ \mathcal{D}$
$h \leftarrow H(m)$
$m' \leftarrow\$ \mathcal{A}(m, h)$
return (m, m')

We have that $\mathbf{Adv}_H^{\mathrm{CR}}(\mathcal{A}') = \mathbf{Adv}_H^{\mathrm{2PRE}}(\mathcal{A})$. Thus if H is collision resistant, H is second pre-image resistant.

$\square$

3.3 Merkle-Damgård Construction

3.4 Construct from compression function

Let k be block length, n be output length, $\mathsf{IV} \in \{0,1\}^n$ be constant. Let $h : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. The Merkle-Damgård Construction is as defined in Figure 17.

Merkle-Damgård(m)

$m' \leftarrow \textsc{pad}(m)$
$m'_1 || \cdots || m'_\ell \leftarrow m'$
$t_0 \leftarrow \mathsf{IV}$
for $i = 1, \cdots, \ell$ **do**
    $t_i \leftarrow h(m'_i, t_{i-1})$
return t_ℓ

Figure 17: Merkle-Damgård Construction

Remarks:

1. Classical construction from block cipher to compression including Davis-Meyer Construction by:
$$h(m_i, t_{i-1}) = E(m_i, t_{i-1}) \oplus t_{i-1}$$
Note that *Davis-Meyer Construction* gives a collision resistant compression function if E is an *ideal cipher*.

3.4.1 Security

Suppose $\textsc{pad}(m)$ transforms m into $m' = m||10^t||[|m|]_L$ where $0 \leq t < k$is minimal such that k divides $|m'|$ and $[\cdot]_L$ denotes L-bit representation of a number where $L \leq K$. If the compression function h is collision-resistant, then so is H.

Proof. Let $\mathcal{A}$ be an adversary against CR of hash function H built from compression function h using the Merkle-Damgård Construction. We construct an adversary $\mathcal{B}$ from $\mathcal{A}$ that breaks CR security of h.

Suppose that $\mathcal{A}$ outputs a colliding pair $X \neq Y$ with non-negligible advantage. Since we know $X \neq Y$, we have that $\textsc{pad}(X)$ and $\textsc{pad}(Y)$ do not need to have the same number of blocks. Let x_i, y_j be their blocks after being padded. We write $\textsc{pad}(X) = x_1, x_2, \cdots, x_u$ and $\textsc{pad}(Y) = y_1, y_2, \cdots, y_v$. Let s_i be the chaining values for X and t_1 be the chaining values for Y. Thus if we look at the last blocks in the two chains, we have that

$$h(s_{u-1}, x_u) = H(X) = H(Y) = h(t_{v-1}, y_v)$$

Now we consider two cases. In the first case, we have that $(s_{u-1}, x_u) \neq (t_{v-1}, y_v)$. In this case, the pair (s_{u-1}, x_u) and (t_{v-1}, y_v) if a collision for h. Then the adversary $\mathcal{B}$ outputs the collision and terminates.

In the second case, we have that we have that $(s_{u-1}, x_u) = (t_{v-1}, y_v)$. Since x_u, y_v both uniquely encode the length of X and Y respectively, we can deduce from $x_u = y_v$ that $u = v$ and the message are of identical length. Now since $s_{u-1} = t_{u-1}$, we have that

$$h(s_{u-2}, x_{u-1}) = s_{u-1} = t_{u-1} = h(t_{v-2}, y_{v-1})$$

We then follow the process and the process must end with a collision in h, otherwise we would eventually find hat all blocks of $\textsc{pad}(X)$ equal those of $\textsc{pad}(Y)$, contradicting the fact that $X \neq Y$. Thus we have that $\mathcal{B}$ must outputs a collision and h.

Therefore, by contraposition, if a compression h is collision resistant, the then hash function constructed from h with Merkle-Damgård Construction is collision resistant. $\square$

3.5 Universal Hashing Function (UHF)

3.5.1 UHF Security

A keyed hash funcion H is an ε-bounded *universal hash function* (ε-UHF) if for any adversary $\mathcal{A}$, the advantage $\mathbf{Adv}_H^{\text{UHF}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}_H^{\text{UHF}}(\mathcal{A}) = \Pr[\textbf{Game } \text{UHF} \Rightarrow 1]$$

Game UHF($\mathcal{A}, H$)

$K \leftarrow\!\!\$\ \mathcal{K}$
$(m_0, m_1) \leftarrow\!\!\$\ \mathcal{A}()$
if $H(K, m_0) = H(K, m_1)$
    $\wedge\, m_0 \neq m_1$ **then**
  return 1
else
  return 0

Figure 18: UHF Game

3.5.2 UHF from Polynomial

Let $\mathbb{F}$ be a finite field, set $\mathcal{K} = \mathcal{T} = \mathbb{F}$, $\mathcal{M} = (\mathbb{F})^{\leq L}$. Define a hash function H_{poly} as:

$$H_{\text{poly}}(K, (a_1, \cdots, a_v)) = K^v + a_1 K^{v-1} + a_2 K^{v-2} + \cdots + a_{v-1} K + a_v \in \mathbb{F}$$

We have that H_{poly} is an ε-UHF for $\varepsilon = \frac{L}{|\mathbb{F}|}$

3.6 Difference Unpredictable Hashing Function (DUHF)

3.6.1 DUHF Security

A keyed hash function H with digest space $\mathcal{T}$ equipped with a group operation "+", is an ε-bounded *difference unpredictable hashing function* if for any adversary $\mathcal{A}$, the advantage $\mathbf{Adv}_H^{\text{DUHF}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}_H^{\text{DUHF}}(\mathcal{A}) = \Pr[\textbf{Game } \text{DUHF} \Rightarrow 1]$$

Game $\mathrm{UHF}(\mathcal{A}, H)$

$K \leftarrow\!\!\$\; \mathcal{K}$

$(m_0, m_1, \delta) \leftarrow\!\!\$\; \mathcal{A}()$

if $H(K, m_0) - H(K, m_1) = \delta$

$\quad\quad \wedge\, m_0 \neq m_1$ **then**

$\quad$ **return** 1

else

$\quad$ **return** 0

Figure 19: DUHF Game

3.6.2 DUHF from Polynomial

Let $\mathbb{F}$ be a finite field, set $\mathcal{K} = \mathcal{T} = \mathbb{F}$, $\mathcal{M} = (\mathbb{F})^{\leq L}$. Define a hash function H_{poly} as:

$$\begin{aligned} H_{\mathrm{Xpoly}}(K, (a_1, \cdots, a_v)) &= K^{v+1} + a_1 K^v + a_2 K^{v-1} + \cdots + a_{v-1} K^2 + a_v K \in \mathbb{F} \\ &= K \cdot H_{\mathrm{poly}}(K, (a_1, \cdots, a_v)) \end{aligned}$$

We have that H_{xpoly} is an ε-UHF for $\varepsilon = \frac{L+1}{|\mathbb{F}|}$

4 Message Authentication Code

4.1 Message Authentication Code (MAC)

A MAC scheme with key space $\mathcal{K}$, message space $\mathcal{M}$ and tag space $\mathcal{T}$ consists of a triple of efficient algorithms $(\text{KGen}, \text{Tag}, \text{Vfy})$ where

$$
\begin{aligned}
&\text{KGen} : \{\} \to \mathcal{K} \\
&\text{Tag} : \mathcal{K} \times \mathcal{M} \to \mathcal{T} \\
&\text{Vfy} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0, 1\}
\end{aligned}
$$

such that

$$\forall K \ \forall m, \text{Vfy}(K, m, \text{Tag}(K, m)) = 1$$

4.2 MAC Unforgeability

4.2.1 EUF-CMA (WUF-CMA) Security

A MAC scheme is $(q_t, q_v, t, \varepsilon)$-*existential unforgeability under chosen message attack* (EUF-CMA) secure, if for any adversaries making q_t queries to tagging oracle OTag, q_v queries to verification OVfy, and running in time at most t, the advantage $\mathbf{Adv}^{\text{EUF-CMA}}_{\text{MAC}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}^{\text{EUF-CMA}}_{\text{MAC}}(\mathcal{A}) = \Pr[\textbf{Game } \text{EUF-CMA} \Rightarrow 1]$$

Game EUF-CMA$(\mathcal{A}, \text{MAC})$

$$
\begin{aligned}
&K \leftarrow\!\!\$\ \text{KGen}(1^\lambda) \\
&\mathcal{Q} \leftarrow \emptyset \\
&(m^*, \tau^*) \leftarrow\!\!\$\ \mathcal{A}^{\text{OTag},\text{OVfy}}() \\
&\textbf{if } m^* \in \mathcal{Q} \textbf{ then} \\
&\quad \textbf{return } 0 \\
&\textbf{else} \\
&\quad b \leftarrow \text{Vfy}_K(m, \tau) \\
&\quad \textbf{return } b
\end{aligned}
$$

Oracle OTag(m)

$$
\begin{aligned}
&\tau \leftarrow \text{Tag}_K(m) \\
&\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\} \\
&\textbf{return } \tau
\end{aligned}
$$

Oracle OVfy(m, τ)

$$
\begin{aligned}
&b \leftarrow \text{Vfy}_K(m, \tau) \\
&\textbf{return } b
\end{aligned}
$$

Figure 20: EUF-CMA Game for MAC

4.2.2 SUF-CMA Security

A MAC scheme is $(q_t, q_v, t, \varepsilon)$-*strong existential unforgeability under chosen message attack* (SUF-CMA) secure, if for any adversaries making q_t queries to tagging oracle OTag, q_v queries to verification oracle OVfy, and running in time at most t, the advantage $\mathbf{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{A}) = \Pr[\textbf{Game } \text{SUF-CMA} \Rightarrow 1]$$

Game SUF-CMA($\mathcal{A}$, MAC)

$K \leftarrow\!\!\$\ \textsc{KGen}(1^\lambda)$
$\mathcal{Q} \leftarrow \emptyset$
$(m^*, \tau^*) \leftarrow\!\!\$\ \mathcal{A}^{\text{OTag},\text{O}\textsc{Vfy}}()$
if $(m^*, \tau^*) \in \mathcal{Q}$ **then**
  return 0
else
  $b \leftarrow \textsc{Vfy}_K(m^*, \tau^*)$
  return b

Oracle OTag(m)

$\tau \leftarrow \textsc{Tag}_K(m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \tau)\}$
return τ

Oracle OVfy(m, τ)

$b \leftarrow \textsc{Vfy}_K(m, \tau)$
return b

Figure 21: SUF-CMA game for MAC

Remarks:

1. EUF-CMA and SUF-CMA security are equivalent if Tag is deterministic and Vfy is built using Tag.
2. For any m and K, there is precisely one value τ for which $\textsc{Vfy}(K, m, \tau) = 1$, so a SUF-CMA adversary does not have more advantage than a EUF-CMA adversary.

4.2.3 No-verify SUF-CMA

Let MAC = (KGen, Tag, Vfy) be a MAC scheme. For any $(q_t, q_v, t, \varepsilon)$-SUF-CMA adversary $\mathcal{A}$ against MAC, there is a $(q_t, t', \varepsilon/q_v)$-no-verify-SUF-CMA advesary $\mathcal{B}$ against MAC with $t' \approx t$.

Game SUF-CMA($\mathcal{A}$, MAC)

$K \leftarrow\!\!\$\ \textsc{KGen}(1^\lambda)$
$\mathcal{Q} \leftarrow \emptyset$
$(m^*, \tau^*) \leftarrow\!\!\$\ \mathcal{A}^{\text{OTag}}()$
if $(m^*, \tau^*) \in \mathcal{Q}$ **then**
  return 0
else
  $b \leftarrow \textsc{Vfy}_K(m, \tau)$
  return b

Oracle OTag(m)

$\tau \leftarrow \textsc{Tag}_K(m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m^*, \tau^*)\}$
return τ

Figure 22: No Verify Oracle SUF-CMA game for MAC

Remarks:

1. This theorem does not hold for EUF-CMA as there are (artifical) MAC schemes which are EUF-CMA secure if $q_t = q$ but there exists an efficient EUF-CMA adversary with advantage 1 if $q_t > 1$
2. The theorem holds if Tag is deterministic and Vfy is built using Tag.

4.3 MACs from PRFs

4.3.1 MACs-from-PRFs Construction

Let $F : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be a pseudorandom function, we build a MAC scheme MAC(F) from F with key space $\mathcal{K}$, message space $\mathcal{M}$, and tag space $\mathcal{T}$ as in Figure 23.

$\textsc{KGen}$

$K \leftarrow\!\$\ \{0,1\}^k$
return K

$\textsc{Tag}(K, m)$

$\tau \leftarrow F(K, m)$
return τ

$\textsc{Vfy}(K, m, \tau)$

$\tau' \leftarrow F(K, m)$
if $\tau = \tau'$ **then**
  return 1
else
  return 0

Figure 23: MAC from PRF construction

4.3.2 MACs-from-PRFs Security

Let $F : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be a function. For any (q_t, t, ε)-SUF-CMA adversary $\mathcal{A}$ against MAC(F), there exists an adversary $\mathcal{B}$ against PRF security of F that runs in time $t' \approx t$, making $q_t + 1$ queries, and has advantage at least $\varepsilon - \frac{1}{|\mathcal{T}|}$.

Proof. Since we have that $\textsc{Tag}$ is deterministic, it suffices to show that if there is an adversary $\mathcal{A}$ against no-verify EUF-CMA security of MAC(F), then there is an adversary $\mathcal{B}$ against PRF security of F with advantage at least $\varepsilon - \frac{1}{|\mathcal{T}|}$. Consider the games G_0 and G_1 in Figure 24. We have that $\mathrm{G}_0 = \mathrm{G}_F^{\text{EUF-CMA}}$ and $\mathrm{G}_1 = \mathrm{G}_f^{\text{EUF-CMA}}$. We define two events W_0 and W_1 where:

- W_0: $\mathcal{A}$ plays G_0 and outputs (m^*, τ^*) such that $\tau^* = F(K, m^*)$ and $m^* \notin \mathcal{Q}$.
- W_1: $\mathcal{A}$ plays G_1 and outputs (m^*, τ^*) such that $\tau^* = f(m^*)$ and $m^* \notin \mathcal{Q}$.

We claim that

$$\begin{aligned}\mathbf{Adv}_F^{\text{EUF-CMA}}(\mathcal{A}) = \Pr[W_0] &= |\Pr[W_0] - \Pr[W_1] + \Pr[W_1]| \\ &\leq |\Pr[W_0] - \Pr[W_1]| + \Pr[W_1]\end{aligned}$$

We construct the adversary $\mathcal{B}$ as in Figure 24. Observe that $\mathcal{B}$ queries its RoR oracle to tag m queried by $\mathcal{A}$, with either the pseudorandom function F or the random function ρ, which simulates the behavior of G_1 or G_2. By the Advantage Rewriting Lemma, we have that

$$\begin{aligned}\mathbf{Adv}_F^{\mathrm{PRF}}(\mathcal{B}) &= \left|\Pr[b' = 0 \mid b = 0] - \Pr[b' = 0 \mid b = 1]\right| \\ &= |\Pr[\tau^* = F(K, m^*) \mid \mathrm{G}_0(\mathcal{A})] - \Pr[\tau^* = f(m^*) \mid \mathrm{G}_1(\mathcal{A})]| \\ &= |\Pr[W_0] - \Pr[W_1]|\end{aligned}$$

We next bound $\Pr[W_1]$. Consider that $\mathcal{A}$ has seen the output of f with input $m_1, m_1, \cdots$ and $\mathcal{A}$ is required to guess the value of f with some new value m^* as input. We have that f is a truly random function, the value of f at m^* is uniformly random and independent from its value on all other inputs. Thus we have that $\Pr[W_1] = \frac{1}{|\mathcal{T}|}$. Therefore, we have that

$$\mathbf{Adv}_F^{\text{EUF-CMA}}(\mathcal{A}) \leq \mathbf{Adv}_F^{\text{PRF}}(\mathcal{B}) + \frac{1}{|\mathcal{T}|}$$

$\square$

Game G_0 G_1

$K \leftarrow\!\!\$\ \textsc{KGen}(1^\lambda)$
$\mathcal{Q} \leftarrow \emptyset$
$(m^*, \tau^*) \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{OTag}}()$
if $m^* \in \mathcal{Q}$ **then**
  return 0
else
  $\tau' \leftarrow F(K, m^*)$
  $\tau' \leftarrow f(m^*)$
return $\tau^* = \tau'$

Oracle $\mathrm{OTag}(m)$

$\tau \leftarrow F(K, m)$
$\tau \leftarrow f(m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
return τ

Oracle $\mathrm{OTag}_{\mathrm{SIM}}(m)$

$\tau \leftarrow \mathrm{RoR}(m)$
return τ

Adversary $\mathcal{B}^{\mathrm{RoR}}$

$(m^*, \tau^*) \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{OTag}_{\mathrm{SIM}}}()$
$\tau' \leftarrow \mathrm{RoR}(m^*)$
if $\tau^* = \tau'$ **then**
  return 0
else
  return 1

Figure 24: Security Proof of MAC construction from PRF

Remark:

(1) This statements implies if F is a PRF, then $\mathrm{MAC}(F)$ is SUF-CMA.

4.4 Domain Extension Theorem

Let $\mathrm{MAC} = (\textsc{KGen}, \textsc{Tag}, \textsc{Vfy})$ be a MAC scheme for message input space $\mathcal{M}$ with tag-length t and key length k. Let $H : \mathcal{M}' \to \mathcal{M}$ be a hash function. Define a new MAC scheme $\textsc{HtMAC} = (\textsc{KGen}, \textsc{Tag}', \textsc{Vfy}')$ for message input space $\mathcal{M}'$ by

- $\textsc{Tag}'(K, m) = \textsc{Tag}(K, H(m))$
- $\textsc{Vfy}'(K, m, \tau) = \textsc{Vfy}(K, H(m))$

For any SUF-CMA adversary $\mathcal{A}$ against HtMAC, we can construct an SUF-CMA adversary $\mathcal{B}$ against MAC, or a collision resistance adversary $\mathcal{C}$ against of H such that

$$\mathbf{Adv}_{\textsc{HtMAC}}^{\mathrm{SUF-CMA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathrm{MAC}}^{\mathrm{SUF-CMA}}(\mathcal{B}) + \mathbf{Adv}_H^{\mathrm{CR}}(\mathcal{C})$$

Proof. Let W_0 denote the event that $\mathcal{A}$ wins SUF-CMA game. Let W_1 denote the event that $H(m) = H(m^*)$ where $m \neq m^*$. We claim that

$$\begin{aligned}\mathbf{Adv}_{\textsc{HtMAC}}^{\mathrm{SUF-CMA}}(\mathcal{A}) &= \Pr[W_0] \\ &= \Pr[W_0 \wedge \neg W_1] + \Pr[W_1 \wedge W_1] \\ &\leq \Pr[W_0 \wedge \neg W_1] + \Pr[W_1]\end{aligned}$$

We first construct the adversary $\mathcal{B}$ as in Figure 25. Observe that in the simulated oracle, $\mathcal{B}$ computes the hash of the message queried by $\mathcal{A}$, and queries its oracle OTag to get the tag, which simulates the SUF-CMA game $\mathcal{A}$ plays. Note that if $\mathcal{A}$ wins the SUF-CMA game, (m^*, τ^*) output by $\mathcal{A}$ has never been queried before. Since in this case, we assume that collision does not happen, thus we have that the hash of m^* has never been queried. Thus if $\mathcal{A}$ wins, we have $\mathcal{B}$ wins, which implies

$$\mathbf{Adv}^{\mathrm{SUF-CMA}}_{\mathrm{HtMAC}}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{SUF-CMA}}_{\mathrm{MAC}}(\mathcal{B})$$

We now construct the adversary C as in Figure 25. Similarly, $\mathcal{C}$ simulates the SUF-CMA game that $\mathcal{A}$ plays. Also, since we assume that collision happens in this case, there must exist some $m' \in \mathcal{Q}$ such that $H(m') = H(m)$ and $m \neq m'$. Thus $\Pr[W_1] \leq \mathbf{Adv}^{\mathrm{CR}}_{H}(\mathcal{C})$.

Finally, we have that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{SUF-CMA}}_{\mathrm{HtMAC}}(\mathcal{A}) &\leq \Pr[W_0 \wedge \neg W_1] + \Pr[W_1] \\ &= \mathbf{Adv}^{\mathrm{SUF-CMA}}_{\mathrm{MAC}}(\mathcal{B}) + \mathbf{Adv}^{\mathrm{CR}}_{H}(\mathcal{C})\end{aligned}$$

□

Adversary $\mathcal{B}^{\mathrm{OTag}}$

$$\begin{array}{l} (m^*, \tau^*) \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{OTag}_{\mathrm{SIM}}}() \\ h^* \leftarrow H(m^*) \\ \textbf{return } (h^*, \tau^*) \end{array}$$

Oracle $\mathrm{OTag}_{\mathrm{SIM}}(m)$

$$\begin{array}{l} h \leftarrow H(m) \\ \tau \leftarrow \mathrm{OTag}(h) \\ \textbf{return } \tau \end{array}$$

Adversary $\mathcal{C}$

$$\begin{array}{l} (X, Y) \leftarrow (\perp, \perp) \\ \mathcal{Q} \leftarrow \emptyset \\ K \leftarrow\!\!\$\ \mathrm{KGen} \\ \mathcal{A}^{\mathrm{OTag}'_{\mathrm{SIM}}} \\ \textbf{return } (X, Y) \end{array}$$

Oracle $\mathrm{OTag}'_{\mathrm{SIM}}(m)$

$$\begin{array}{l} h \leftarrow H(m) \\ \textbf{if } \exists m' \in \mathcal{Q} : H(m') = h \\ \qquad \wedge\, m \neq m' \textbf{ then} \\ \quad (X, Y) \leftarrow (m, m') \\ \tau \leftarrow\!\!\$\ \mathrm{Tag}(K, h) \\ \mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\} \\ \textbf{return } \tau \end{array}$$

Figure 25: Adversary $\mathcal{B}$ and $\mathcal{C}$ for proof of Domain Extension Theorem

4.5 Nonce-based MACs

4.5.1 NMAC

A nonce-based MAC scheme with key space $\mathcal{K}$, nonce space $\mathcal{N}$ and tag space $\mathcal{T}$, consists of a triple of efficient algorithms (KGen, Tag, Vfy) where

$$\begin{aligned} &\mathrm{KGen} : \{\} \to \mathcal{K} \\ &\mathrm{Tag} : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \to \mathcal{T} \\ &\mathrm{Vfy} : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \times \mathcal{T} \to \{0, 1\} \end{aligned}$$

such that

$$\forall K \in \mathcal{K}\ \forall N \in \mathcal{N}\ \forall m \in \mathcal{M}, \textsc{Vfy}(K, N, m, \textsc{Tag}(K, N, m))$$

4.5.2 SUF-CMA Security of NMAC

A nonce-based MAC scheme is $(q_t, q_v, t, \varepsilon)$-SUF-CMA secure if for all adversaries $\mathcal{A}$ running in time at most t, making at most q_t tagging queries and at most q_v verification queries, the advantage $\mathbf{Adv}^{\text{SUF-CMA}}_{\text{NMAC}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}^{\text{SUF-CMA}}_{\text{NMAC}}(\mathcal{A}) = \Pr[\textbf{Game}\ \text{SUF-CMA} \Rightarrow 1]$$

Game SUF-CMA($\mathcal{A}$, MAC)

$K \leftarrow\!\!\$\ \textsc{KGen}(1^\lambda)$
$\mathcal{Q} \leftarrow \emptyset$
$(N^*, m^*, \tau^*) \leftarrow\!\!\$\ \mathcal{A}^{\text{OTag,OVfy}}()$
if $(N^*, m^*, \tau^*) \in \mathcal{Q}$ **then**
  return 0
else
  $b \leftarrow \textsc{Vfy}_K(N, m, \tau)$
  return b

Oracle OTag(N, m)

$\tau \leftarrow \textsc{Tag}_K(N, m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(N, m, \tau)\}$
return τ

Oracle OVfy(N, m, τ)

$b \leftarrow \textsc{Vfy}_K(N, m, \tau)$
return b

Figure 26: SUF-CMA game for NMAC

4.6 UHF-then-PRF Composition

4.6.1 Compose UHF and PRF

Let H be an ε-UHF with key space $\mathcal{K}$, message space $\mathcal{M}$ and digest space $\mathcal{T}$. Let F be a secure PRF with key space $\mathcal{K}'$, message space $\mathcal{T}$ and output space $\mathcal{X}$. Define a function F' by

$$F'((K_1, K_2), m) := F(K_2, H(K_1, m))$$

Then F' is a secure PRF with key space $\mathcal{K} \times \mathcal{K}'$, message space $\mathcal{M}$ and output space $\mathcal{X}$.

4.6.2 UHF-then-PRF Composition Security

Let $\mathcal{A}$ be a PRF adversary against F' making at most q queries, then there exists a PRF adversary $\mathcal{B}$ against F making q queries such that

$$\mathbf{Adv}^{\text{PRF}}_{F'}(\mathcal{A}) \leq \mathbf{Adv}^{\text{PRF}}_{F}(\mathcal{B}) + \frac{q^2}{2} \cdot \varepsilon$$

Proof. Let $\mathcal{A}$ be a PRF adversary against F', we construct a PRF adversary $\mathcal{B}$ against F as in Figure 28. Observe that $\mathcal{B}$ makes the same number of queries as $\mathcal{A}$ does, also $\mathcal{B}$ first hashes the query from $\mathcal{A}$ and then queries the hashes with its oracle RoR, which simulates the PRF game that $\mathcal{A}$ plays. Also, $\mathcal{B}$ runs in essentially the same time as $\mathcal{A}$. Thus $\mathcal{B}$ perfectly simulates the PRF game of $\mathcal{A}$. Observe that $\mathcal{B}$ returns the same bit as $\mathcal{A}$. Thus if $\mathcal{A}$ wins the game, then $\mathcal{B}$ wins the game.

In the second case, we can construct a UHF adversary $\mathcal{D}$ against H as in Figure 27. Since ρ is a random function, if we have that $f(H(K_1, m)) = f(H(K_2, m'))$ for $m \neq m'$, then $\mathcal{D}$ wins the UHF game. Since $\mathcal{A}$ makes q queries, there are $\frac{q(q-1)}{2}$ pairs of indices.

By Union Bound, we have that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{PRF}}_{F'}(\mathcal{A}) &\leq \mathbf{Adv}^{\mathrm{PRF}}_{F}(\mathcal{B}) + \frac{q(q-1)}{2} \cdot \varepsilon \\ &\leq \mathbf{Adv}^{\mathrm{PRF}}_{F}(\mathcal{B}) + \frac{q^2}{2} \cdot \varepsilon\end{aligned}$$

□

Adversary $\mathcal{B}^{\mathrm{RoR}}$	**Oracle** $\mathrm{RoR}_{\mathrm{SIM}}(m)$
$K_1 \leftarrow\!\!\$\ \mathcal{K}$	$h \leftarrow H(K_1, m)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{RoR}_{\mathrm{SIM}}}()$	$c \leftarrow \mathrm{RoR}(h)$
return b'	**return** c

Figure 27: Adversary $\mathcal{B}$ for UHF-PRF Construction

Adversary $\mathcal{D}$	**Oracle** $\mathrm{OIdeal}(m)$
$(X, Y) \leftarrow (\perp, \perp)$	$h \leftarrow H(K_1, m)$
$\mathcal{Q} \leftarrow$	**if** $\exists m' \in \mathcal{Q}$:
$K_1 \leftarrow\!\!\$\ \mathcal{K}$	$\quad m \neq m' \wedge h = H(K_1, m')$ **then**
$\rho \leftarrow\!\!\$\ \mathcal{F}[\mathcal{T}]$	$\quad (X, Y) \leftarrow (m, m')$
$\mathcal{A}^{\mathrm{OIdeal}}()$	$c \leftarrow \rho(h)$
return (X, Y)	**return** c

Figure 28: Adversary $\mathcal{D}$ for UHF-PRF Construction

4.7 Carter-Wegman (CW) MAC

4.7.1 CW-MAC Construction

Let H be a ε-DUHF with outputs in $\mathcal{T}_H$; Let F be a PRF on $\{0,1\}^n$ with output in $\mathcal{T}_H$; assume that $(\mathcal{T}_H, +)$ is a group, define CW-MAC(F, H) as follows:

$\textsc{KGen}(1^\lambda)$	$\textsc{Tag}((K_1, K_2), N, m)$
$(K_1, K_2) \leftarrow\!\!\$\; \mathcal{K}_H \times \mathcal{K}_F$	$\tau \leftarrow H(K_1, m) + F(K_2, N)$
return (K_1, K_2)	**return** τ
	$\textsc{Vfy}((K_1, K_2), N, m, \tau)$
	$\tau' \leftarrow \textsc{Tag}((K_1, K_2), N, m)$
	return $\tau = \tau'$

Figure 29: CW-MAC Construction

4.7.2 CW-MAC Security

For any SUF-CMA adversary $\mathcal{A}$ against CW-MAC(F, H) making q_t tag queries, there exists a PRF adversary $\mathcal{B}$ against F such that

$$\mathbf{Adv}^{\text{SUF-CMA}}_{\text{CW-MAC}(F,H)}(\mathcal{A}) \leq \mathbf{Adv}^{\text{PRF}}_{F}(\mathcal{B}) + \varepsilon + \frac{1}{|\mathcal{T}_H|}$$

Proof. Since we have that $\textsc{Tag}$ is deterministic, it suffices to show the no-verify EUF-CMA security. Define G_0 and G_1 as in Figure 30. Let W_i be the event that $\mathcal{A}$ wins in game G_i respectively. We have that

$$\mathbf{Adv}^{\text{SUF-CMA}}_{\text{CW-MAC}(F,H)}(\mathcal{A}) \leq |\Pr[W_0] - \Pr[W_1]| + \Pr[W_1]$$

We construct a PRP adversary $\mathcal{B}$ against F as in Figure 30. Observe that $\mathcal{B}$ makes the same number of queries as $\mathcal{A}$, and $\mathcal{B}$ samples the a hash key and run $H(K_1, m)$ with m from $\mathcal{A}$, queries its oracle RoR with the nonce queried by $\mathcal{A}$, and then output the tag after group operation, which simulates the SUF-CMA game that $\mathcal{A}$ plays. By Advantage Rewriting Lemma, we have that

$$\begin{aligned}\mathbf{Adv}^{\text{PRF}}_{F}(\mathcal{B}) &= \big|\Pr[b' = 0 \mid b = 0] - \Pr[b' = 0 \mid b = 1]\big| \\ &= |\Pr[W_0] - \Pr[W_1]|\end{aligned}$$

We then show that $\Pr[W_1] \leq \varepsilon + \frac{1}{|\mathcal{T}_H|}$. Let E_1 denote the event that $\mathcal{A}$ wins and output a triple (N^*, m^*, τ^*) in which N^* has neven been used in any of $\mathcal{A}$'s tag queries. Let E_2 denote the event that $\mathcal{A}$ wins and output a triple (N^*, m^*, τ^*) in which $N^* = N$ with N repeated from some previous tag query. We claim that

$$\Pr[W_1] = \Pr[E_1] + \Pr[E_2]$$

In E_1, for $\mathcal{A}$ to win, we must have $\tau^* = H(K_1, m^*) + f(N^*)$. Note that after rearranging, we have that $f(N^*)$ is a group element in $\mathcal{T}_H$. Since N^* is new, $f(N^*)$ is uniformly random in $\mathcal{T}_H$ and independent from all the other outputs of f seen by $\mathcal{A}$. Thus we have that

$$\Pr[E_1] = \frac{1}{|\mathcal{T}_H|}$$

In E_2, we then have $\tau^* = H(K_1, m^*) + f(N)$ and $\tau = H(K_1, m) + f(N)$ for some N. Thus we have that $\tau^* - \tau = H(K_1, m^*) - H(K_1, m)$. We can then build an adversary $\mathcal{D}$ that breaks DUHF security of H with output $(m^*, m, \tau^* - \tau)$. Thus we have that

$$\Pr[E_2] \leq \mathbf{Adv}_H^{\mathrm{DUHF}}(\mathcal{D}) \leq \varepsilon$$

Finally, we have that

$$\begin{aligned}\mathbf{Adv}_{\text{CW-MAC}(F,H)}^{\text{SUF-CMA}}(\mathcal{A}) &\leq |\Pr[W_0] - \Pr[W_1]| + \Pr[W_1] \\ &= \mathbf{Adv}_F^{\mathrm{PRP}}(\mathcal{B}) + \varepsilon + \frac{1}{|\mathcal{T}_H|}\end{aligned}$$

□

Game G_0 G_1

$(K_1, K_2) \leftarrow\$ \ \mathrm{KGen}(1^\lambda)$
$\rho \leftarrow\$ \ \mathcal{F}[\{0,1\}^n]$
$\mathcal{Q} \leftarrow \emptyset$
$(N^*, m^*, \tau^*) \leftarrow\$ \ \mathcal{A}^{\mathrm{OTag}}()$
if $m^* \in \mathcal{Q}$ **then**
  return 0
else
  $\tau' \leftarrow H(K_1, m^*) + F(K_2, N^*)$
  $\tau' \leftarrow H(K_1, m^*) + f(N^*)$
return $\tau^* = \tau'$

Oracle $\mathrm{OTag}(N, m)$

$\tau \leftarrow H(K_1, m) + F(K, N)$
$\tau \leftarrow H(K_1, m) + f(N)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
return τ

Oracle $\mathrm{OTag}_{\mathrm{SIM}}(N, m)$

$c \leftarrow \mathrm{RoR}(N)$
$\tau \leftarrow H(K_1, m) + c$
return τ

Adversary $\mathcal{B}^{\mathrm{RoR}}$

$K_1 \leftarrow\$ \ \mathcal{K}_H$
$(N^*, m^*, \tau^*) \leftarrow\$ \ \mathcal{A}^{\mathrm{OTag}_{\mathrm{SIM}}}()$
$c' \leftarrow \mathrm{RoR}(m^*)$
$\tau' \leftarrow H(K_1, m^*) + c'$
if $\tau^* = \tau'$ **then**
  return 0
else
  return 1

Figure 30: Security Proof of MAC construction from PRF

5 Asymmetric Encryption

5.1 Public Key Encryption

5.1.1 Public Key Encryption Scheme

A *public key encryption* scheme PKE with public key space $\mathcal{PK}$, secret key space $\mathcal{SK}$, message space $\mathcal{M}$, and ciphertext space $\mathcal{C}$, consists of a triple of efficient algorithms PKE = $(\text{KG}\textsc{en}, \textsc{Enc}, \textsc{Dec})$ where

$$\begin{aligned}
&\text{KG}\textsc{en} : \{\} \to \mathcal{PK} \times \mathcal{SK} \\
&\textsc{Enc} : \mathcal{PK} \times \mathcal{M} \to \mathcal{C} \\
&\textsc{Dec} : \mathcal{SK} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}
\end{aligned}$$

such that

$$\forall (\mathsf{pk}, \mathsf{sk}) \in \mathcal{PK} \times \mathcal{SK}\ \forall m \in \mathcal{M}, \textsc{Dec}(\mathsf{sk}, \textsc{Enc}(\mathsf{pk}, m)) = m$$

5.1.2 IND-CCA security of PKE

A public key encryption scheme PKE is defined to be $(q_e, q_d, t, \varepsilon)$-*indistinguishibility under chosen ciphertext attack* (IND-CCA), if for any adversaries $\mathcal{A}$ running in time at most t and making at most q_e encryption queries to oracle LoR and at most q_d decryption queries to oracle ODec, the advantage $\mathbf{Adv}^{\text{IND-CPA}}_{\text{PKE}}(\mathcal{A}) \leq \varepsilon$.

$$\mathbf{Adv}^{\text{IND-CCA}}_{\text{PKE}}(\mathcal{A}) = 2 \cdot |\Pr[\textbf{Game } \text{IND-CCA}(\mathcal{A}, \text{SE}) \Rightarrow \mathsf{true}] - \frac{1}{2}|$$

Game IND-CCA($\mathcal{A}$, PKE)	**Oracle** LoR(m_0, m_1)	**Oracle** ODec(c)
$b \leftarrow\!\!\$\ \{0,1\}$	**if** $\lvert m_0 \rvert \neq \lvert m_1 \rvert$ **then**	**if** $c \in \mathcal{Q}$ **then**
$\mathsf{pk}, \mathsf{sk} \leftarrow\!\!\$\ \text{KG}\textsc{en}(1^\lambda)$	**return** $\perp$	**return** $\perp$
$\mathcal{Q} \leftarrow \emptyset$	$c \leftarrow\!\!\$\ \textsc{Enc}(\mathsf{pk}, m_b)$	$m \leftarrow \textsc{Dec}(\mathsf{sk}, c)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{LoR,ODec}}(\mathsf{pk})$	$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{c\}$	**return** m
return $b' = b$	**return** c	

Figure 31: IND-CCA Game of a Public Key Encryption Scheme

5.2 KEM and DEM

5.2.1 Key Encapsulation Mechanism

A *key encapsulation mechanism* KEM with public key space $\mathcal{PK}$, secret key space $\mathcal{SK}$, symmetric key space $\mathcal{K}$, and encapsulation space $\mathcal{C}$, consists of a triple of efficient algorithms KEM = $(\text{KG}\textsc{en}, \textsc{Encap}, \textsc{Decap})$ where

$$\begin{aligned}
&\text{KG}\textsc{en} : \{\} \to \mathcal{SK} \times \mathcal{PK} \\
&\textsc{Encap} : \mathcal{PK} \to \mathcal{C} \times \mathcal{K} \\
&\textsc{Decap} : \mathcal{SK} \times \mathcal{C} \to \mathcal{K} \cup \{\perp\}
\end{aligned}$$

such that

$$\forall(\mathsf{sk},\mathsf{pk}) \in \mathcal{SK} \times \mathcal{PK}, \textsc{Encap}(\mathsf{pk}) = (c, K) \Rightarrow K = \textsc{Decap}(\mathsf{sk}, c)$$

5.2.2 IND-CCA Security for KEM

A key encapsulation mechanism KEM is defined to be $(q_e, q_d, t, \varepsilon)$-*indistinguishibility under chosen ciphertext attack* (IND-CCA), if for any adversaries $\mathcal{A}$ running in time at most t and making at most q_e encryption queries to oracle LoR and at most q_d decryption queries to oracle ODec, the advantage $\mathbf{Adv}^{\text{IND-CPA}}_{\text{KEM}}(\mathcal{A}) \leq \varepsilon$.

$$\mathbf{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{A}) = 2 \cdot |\Pr[\mathbf{Game}\ \text{IND-CCA}(\mathcal{A}, \text{KEM}) \Rightarrow \mathsf{true}] - \frac{1}{2}|$$

Game IND-CCA($\mathcal{A}$, KEM)	**Oracle** ODec(c)
1 : $b \leftarrow\$ \{0,1\}$	1 : **if** $c = c_0$ **then**
2 : $\mathsf{pk}, \mathsf{sk} \leftarrow\$ \textsc{KGen}(1^\lambda)$	2 : **return** $\perp$
3 : $(c_0, K_0) \leftarrow\$ \textsc{Encap}(\mathsf{pk})$	3 : $K \leftarrow \textsc{Decap}(\mathsf{sk}, c)$
4 : $K_1 \leftarrow\$ \mathcal{K}$	4 : **return** K
5 : $b' \leftarrow\$ \mathcal{A}^{\text{ODec}}(\mathsf{pk}, c_0, K_b)$	
6 : **return** $b' = b$	

Figure 32: IND-CCA Game of a Public Key Encryption Scheme

5.2.3 KEM/DEM Composition

Let KEM = (KGen, Encap, Decap), and DEM = (KGen, Enc, Dec) be a DEM such that KEM.$\mathcal{K}$ = DEM.$\mathcal{K}$, then we build a PKE scheme PKE = (KGen, Enc, Dec) from KEM and DEM as in Figure 33.

PKE.KGen	PKE.Enc(m)	PKE.Dec(sk, c)
1 : $\mathsf{sk}, \mathsf{pk} \leftarrow\$$ KEM.KGen	1 : $(c_K, K) \leftarrow\$$ KEM.Encap(pk)	1 : $c_K \| c_m \leftarrow c$
2 : **return** $(\mathsf{sk}, \mathsf{pk})$	2 : $c_m \leftarrow\$$ DEM.Enc(K, m)	2 : $K \leftarrow$ KEM.Decap(sk, c_K)
	3 : **return** $c_K \| c_m$	3 : **if** $K = \perp$ **then**
		4 : **return** $\perp$
		5 : $m \leftarrow$ DEM.Dec(K, c_m)
		6 : **return** m

Figure 33: KEM/DEM Composition

5.2.4 Security of KEM/DEM Composition

For any 1-query IND-CCA adversary $\mathcal{A}$ against PKE from KEM/DEM composition, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that

$$\mathbf{Adv}^{\text{IND-CCA}}_{\text{PKE}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\text{IND-CCA}}_{\text{KEM}}(\mathcal{B}) + \mathbf{Adv}^{\text{IND-CCA}}_{\text{DEM}}(\mathcal{C})$$

5.3 RSA Encryption

5.3.1 Textbook RSA

Define the textbook RSA cryptosystem as in Figure 34.

$\text{KG}\textsc{en}(\ell)$

$p, q \leftarrow\!\!\$\ \mathsf{Prime}(\ell/2)$
    // p, q of bit-size $\ell/2$
$N \leftarrow p \cdot q$
$d \leftarrow\!\!\$\ \mathbb{Z}_N^*$
$e \leftarrow d^{-1} \bmod \phi(N)$
$\mathsf{pk} \leftarrow (e, N)$
$\mathsf{sk} \leftarrow d$
return $(\mathsf{pk}, \mathsf{sk})$

$\text{E}\textsc{nc}(\mathsf{pk}, m)$

$(e, N) \leftarrow \mathsf{pk}$
$c \leftarrow m^e \bmod N$
return c

$\text{D}\textsc{ec}(\mathsf{sk}, c)$

$d \leftarrow \mathsf{sk}$
$m \leftarrow c^d \bmod N$
return m

Figure 34: Textbook RSA

By Euler's Theorem, the correctness is defined by:

$$(m^e)^d \equiv m^{k\cdot\phi(N)+1} \equiv m^{k\cdot\phi(N)} \cdot m \equiv m \pmod{N}$$

5.3.2 RSA inversion Problem

Define the *RSA Inversion Problem* as in Figure 35.

Game $\text{RSAI}\textsc{nv}(\mathcal{A})$

$\mathsf{sk}, \mathsf{pk} \leftarrow\!\!\$\ \text{RSA.KG}\textsc{en}$
$d \leftarrow \mathsf{sk}$
$e, N \leftarrow \mathsf{pk}$
$x \leftarrow\!\!\$\ \mathbb{Z}_N$
$y \leftarrow x^e \bmod N$
$x' \leftarrow \mathcal{A}(N, e, y)$
return $x = x'$

Figure 35: RSA Inversion Problem

Remarks:

(1) If $\mathcal{A}$ can factor N, then $\mathcal{A}$ can solve the RSA inversion problem.

(2) The reverse implicaion is open, but no algorithm faster than factoring N is known for solving RSA inversion in general.

5.3.3 Build KEM from RSA

Let $H : \mathbb{Z}_N \to \{0,1\}^k$ be a hash function. We can build a KEM from RSA as in Figure 36.

$\textsc{KGen}(\ell)$	$\textsc{Encap}(\mathsf{pk}, m)$
$p, q \leftarrow\!\$\ \mathsf{Prime}(\ell/2)$	$(e, N) \leftarrow \mathsf{pk}$
$/\!/\ p, q$ of bit-size $\ell/2$	$s \leftarrow \mathbb{Z}_N$
$N \leftarrow p \cdot q$	$c \leftarrow s^e \bmod N$
$d \leftarrow\!\$\ \mathbb{Z}_N^*$	$K \leftarrow H(s)$
$e \leftarrow d^{-1} \bmod \phi(N)$	**return** (c, K)
$\mathsf{pk} \leftarrow (e, N)$	$\textsc{Decap}(\mathsf{sk}, c)$
$\mathsf{sk} \leftarrow d$	$d \leftarrow \mathsf{sk}$
return $(\mathsf{pk}, \mathsf{sk})$	$s \leftarrow c^d \bmod N$
	$K \leftarrow H(s)$
	return K

Figure 36: Build KEM from RSA

Remarks:

1. RSA-KEM is IND-CCA secure under Random Oracle Model (ROM) provided RSA inversion problem is hard.

5.4 Discrete Log Setting

5.4.1 DLog Problem

Let p, q be primes such that $p = kq + 1$ for some $k \in \mathbb{Z}^+$. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$ such that $\mathbb{G} = \langle g \rangle$ for some geneator g and $|\mathbb{G}| = q$. Define the *discrete log problem* (DLP) as in Figure 37.

Game $\textsc{DLog}(\mathcal{A})$
$x \leftarrow\!\$\ \mathbb{Z}_q$
$x' \leftarrow \mathcal{A}(g, g^x)$
return $x = x'$

Figure 37: Discrete Log Problem

5.4.2 CDH Problem

Let p, q be primes such that $p = kq+1$ for some $k \in \mathbb{Z}^+$. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$ such that $\mathbb{G} = \langle g \rangle$ for some geneator g and $|\mathbb{G}| = q$. Define the *computational Diffie-Hellman problem* (CDH) as in Figure 38.

Game CDH($\mathcal{A}$)

1 :	$x, y \leftarrow\!\!\$\ \mathbb{Z}_q$
2 :	$Z \leftarrow \mathcal{A}(g, g^x, g^y)$
3 :	**return** $Z = g^{ab}$

Figure 38: Computational Diffie-Hellman Problem

5.4.3 DDH Problem

Let p, q be primes such that $p = kq + 1$ for some $k \in \mathbb{Z}^+$. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$ such that $\mathbb{G} = \langle g \rangle$ for some geneator g and $|\mathbb{G}| = q$. Define the *Decisional Diffie-Hellman problem* (DDH) as in Figure 39.

Game DDH($\mathcal{A}$)

1 :	$b \leftarrow\!\!\$\ \{0, 1\}$
2 :	$x, y, z \leftarrow\!\!\$\ \mathbb{Z}_q$
3 :	$Z_0 \leftarrow g^{ab}$
4 :	$Z_1 \leftarrow g^c$
5 :	$b' \leftarrow \mathcal{A}(g, g^x, g^y, Z_b)$
6 :	**return** $b = b'$

Figure 39: Decisional Diffie-Hellman Problem

5.5 Diffie-Hellman Key Exchange

Let p, q be primes such that $p = kq + 1$ for some $k \in \mathbb{Z}^+$. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$ such that $\mathbb{G} = \langle g \rangle$ for some geneator g and $|\mathbb{G}| = q$. Define the *Diffie-Hellman Key Exchange* as in Figure 40.

Alice		**Bob**
$a \leftarrow\!\$\ \mathbb{Z}_q$		
$K_a \leftarrow g^a$	$\xrightarrow{K_a}$	
		if $K_a^q \neq 1$ **then**
		return $\perp$
		$b \leftarrow\!\$\ \mathbb{Z}_q$
	$\xleftarrow{K_b}$	$K_b \leftarrow g^b$
if $K_b^q \neq 1$ **then**		
return $\perp$		
$K \leftarrow \mathrm{KDF}(K_b^a)$		$K \leftarrow \mathrm{KDF}(K_a^b)$

Figure 40: Diffie-Hellman Key Exchange

5.6 ElGamal Encryption

Let p, q be primes such that $p = kq + 1$ for some $k \in \mathbb{Z}^+$. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$ such that $\mathbb{G} = \langle g \rangle$ for some geneator g and $|\mathbb{G}| = q$. Define the *ElGamal Public-Key Encryption Scheme* as in Figure 41

$\text{KGen}(\ell)$	$\text{Enc}(\mathsf{pk}, M)$	$\text{Dec}(\mathsf{sk}, R, C)$
$x \leftarrow\!\$\ \mathbb{Z}_q$	$X \leftarrow \mathsf{pk}$	$x \leftarrow \mathsf{sk}$
$X \leftarrow g^x$	$r \leftarrow\!\$\ \mathbb{Z}_q$	**if** $R^q \neq 1$ **then**
$\mathsf{pk} \leftarrow X$	$R \leftarrow g^r$	**return** $\perp$
$\mathsf{sk} \leftarrow x$	$Z \leftarrow X^r$	$Z \leftarrow R^x$
	$C \leftarrow M \cdot Z$	$M \leftarrow C \cdot Z^{-1}$
	return (R, C)	**return** M

Figure 41: ElGamal Public Key Encryption

The correctness is defined by:

$$M \cdot X^r \cdot R^{-x} = M \cdot g^{xr} \cdot g^{-rx} = M$$

5.7 DHIES

Let p, q be primes such that $p = kq + 1$ for some $k \in \mathbb{Z}^+$. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$ such that $\mathbb{G} = \langle g \rangle$ for some geneator g and $|\mathbb{G}| = q$. Let H be a hash function with suitable output domain. Let AE be an authenticated encryption scheme. Define the *Diffie-Hellman Integrated Encryption Scheme* (DHIES) as in Figure 42.

$\textsc{KGen}(\ell)$	$\textsc{Enc}(\mathsf{pk}, M)$	$\textsc{Dec}(\mathsf{sk}, R, C)$
$x \leftarrow\$\ \mathbb{Z}_q$	$X \leftarrow \mathsf{pk}$	$x \leftarrow \mathsf{sk}$
$X \leftarrow g^x$	$r \leftarrow\$\ \mathbb{Z}_q$	**if** $R^q \neq 1$ **then**
$\mathsf{pk} \leftarrow X$	$R \leftarrow g^r$	**return** $\perp$
$\mathsf{sk} \leftarrow x$	$Z \leftarrow X^r$	$X \leftarrow g^x$
	$K \leftarrow H(X, R, Z)$	$Z \leftarrow R^x$
	$K_e, K_m \leftarrow K$	$K \leftarrow H(X, R, Z)$
	$C \leftarrow \mathrm{AE}.\textsc{Enc}(K_e, K_m, M)$	$K_e, K_m \leftarrow K$
	return (R, C)	$M \leftarrow \mathrm{AE}.\textsc{Dec}(K_e, K_m, C)$
		return M

Figure 42: Diffie-Hellman Intergrated Encryption Scheme

Remarks:

1. DHIES is IND-CCA secure under Random Oracle Model.

6 Digital Signature

6.1 Digital Signature Scheme

A signature scheme SIG with signing key space $\mathcal{SK}$, verification key space $\mathcal{VK}$, message space $\mathcal{M}$, and signature space Σ consists of a triple algorithm (KGEN, SIG, VFY) where

$$\begin{aligned}&\textsc{KGen} : \{\} \to \mathcal{SK} \times \mathcal{VK}\\&\textsc{Sig} : \mathcal{SK} \times \mathcal{M} \to \Sigma\\&\textsc{Vfy} : \mathcal{VK} \times \Sigma \times \mathcal{M} \to \{0,1\}\end{aligned}$$

such that

$$\forall m \in \mathcal{M}\ \forall (sk, vk) \in \mathcal{SK} \times \mathcal{VK}, \textsc{Vfy}(vk, \textsc{Sig}(sk, m), m) = 1$$

6.2 Signature Unforgeability

6.2.1 EUF-CMA Security

A signature scheme is (q_s, t, ε)-*existential unforgeability under chosen message attack* (EUF-CMA) secure, if for any adversaries making q_s queries to signing oracle OSig, and running in time at most t, the advantage $\mathbf{Adv}^{\text{EUF-CMA}}_{\text{MAC}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}^{\text{EUF-CMA}}_{\textsc{Sig}}(\mathcal{A}) = \Pr[\mathbf{Game}\ \text{EUF-CMA}(\textsc{Sig}, \mathcal{A}) \Rightarrow 1]$$

Game EUF-CMA($\mathcal{A}$, SIG)	**Oracle** OSig(m)
$vk, sk \leftarrow\$ \textsc{KGen}(1^\lambda)$	$\sigma \leftarrow \textsc{Sig}(sk, m)$
$\mathcal{Q} \leftarrow \emptyset$	$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
$(m^*, \sigma^*) \leftarrow\$ \mathcal{A}^{\text{OSig}}()$	**return** σ
if $m^* \in \mathcal{Q}$ **then**	
return 0	
else	
$b \leftarrow \textsc{Vfy}(pk, m, \sigma)$	
return b	

Figure 43: EUF-CMA Game for SIG

6.2.2 SUF-CMA Security

A signature scheme is (q_s, t, ε)-*strong existential unforgeability under chosen message attack* (SUF-CMA) secure, if for any adversaries making q_s queries to signing oracle OSig, and running in time at most t, the advantage $\mathbf{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{A}) \leq \varepsilon$ where

$$\mathbf{Adv}^{\text{SUF-CMA}}_{\textsc{Sig}}(\mathcal{A}) = \Pr[\mathbf{Game}\ \text{SUF-CMA}(\textsc{Sig}, \mathcal{A}) \Rightarrow 1]$$

Game SUF-CMA($\mathcal{A}$, SIG)

$vk, sk \leftarrow\!\$\ \textsc{KGen}(1^\lambda)$
$\mathcal{Q} \leftarrow \emptyset$
$(m^*, \sigma^*) \leftarrow\!\$\ \mathcal{A}^{\mathrm{OSig}}()$
if $(m^*, \sigma^*) \in \mathcal{Q}$ **then**
  return 0
else
  $b \leftarrow \textsc{Vfy}(pk, m, \sigma)$
  return b

Oracle OSig(m)

$\sigma \leftarrow \textsc{Sig}(sk, m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m, \sigma\}$
return σ

Figure 44: SUF-CMA Game for SIG