Lecture note 2: Symmetric key cryptography

Haiyang Xue

January 25, 2023

In this lecture, we discuss the syntax of symmetric key encryption, its security and constructions. As important building blocks, pseudorandom generator (PRG), pseudurandom function (PRF), and message authenticated code (MAC) are also included. At last, we introduce the concept of hash functions.

1 Syntax of symmetric key encryption

A symmetric key encryption scheme Π consists of three public algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, as well as message space $\mathcal{M}$, key space $\mathcal{K}$, and ciphertext speace $\mathcal{C}$.

$\mathsf{KeyGen}\ (1^\lambda)$ On input 1^λ, generate $K \leftarrow K$ with randomness, and output a pair key (K, K) as secret keys. Distribute the key to the two parties.

$\mathsf{Enc}(K, M)$ This is a probabilistic or deterministic algorithm. On input secret key K and message $M \in \mathcal{M}$, generate and output $C \in \mathcal{C}$ as the ciphertext.

$\mathsf{Dec}(K, C)$ This is a deterministic algorithm. On input secret key K and the ciphertext $C \in \mathcal{C}$, return the corresponding message $M \in \mathcal{M}$ or $\perp$.

Decryption correctness requires that for all $(K, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $M = \mathsf{Dec}(K, \mathsf{Enc}(K, M))$ holds for $m \in \mathcal{M}$. This is known as the perfect correctness. Sometimes, "a small" probability of decryption error is allowed.

Remark 1. *All the algorithms in the encryption scheme are public. The only thing that is secret is the encryption/decrytion key K. This is known as the Kerckhoffs' principle.*

Remark 2. *We leave the problem of distributing/sharing secret key K in the next lecture.*

2 Perfect security and one-time pad

Generally, if an encryption is secure against any unbounded adversary, it satisfies perfect security. Informally, the ciphertext gives nothing about the message. Shannon formally defined perfect security as follows.

Definition 1. *We say that an encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *associated with message space* $\mathcal{M}$ *is perfectly secret, if for every distribution over* $\mathcal{M}$*, every* $m \in \mathcal{M}$*, and every ciphertext* $c \in \mathcal{C}$*,*

$$\Pr[M = m \mid C = c] = \Pr[M = m], \tag{1}$$

with probability taken over the random choice in KeyGen*, and the random coins used by* Enc*.*

A question is whether perfect security is achievable. One-time padding scheme $\Pi_{\mathsf{otp}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, introduced in the following, achieves the perfect security. Fix an integer l and set $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^l$.

$\mathsf{KeyGen}\ (1^\lambda)$ On input 1^λ, generate $K \leftarrow \{0,1\}^l$ and output a pair key (K, K) as secret keys.

$\mathsf{Enc}(K, M)$ On input secret key K and message M, generate and output $C = K \oplus M$ as the ciphertext.

$\mathsf{Dec}(K, C)$ On input secret key K and the ciphertext C, return $M = K \oplus C$ as the message.

Theorem 1. *One-time pad scheme* Π_{otp} *is perfectly secret.*

Proof. We only need to prove that the one-time padding satisfies equation 1.

At first we have,

$$\begin{aligned} \Pr[C = c \mid M = m] &= \Pr[m = c \oplus K] \\ &= \Pr[K = m \oplus c] \\ &= \Pr[K = c \oplus m] \\ &= \frac{1}{2^l}. \end{aligned} \tag{2}$$

Then, we also have

$$\begin{aligned} \Pr[C = c] &= \sum_{m \in \{0,1\}^l} \Pr[C = c | M = m] \Pr[M = m] \\ &= \sum_{m \in \{0,1\}^l} \frac{1}{2^l} \Pr[M = m] \quad \text{according to equation 2} \\ &= \frac{1}{2^l}. \end{aligned} \tag{3}$$

Finally, according to Bayes theorem,

$$\begin{aligned} \Pr[M = m \mid C = c] &= \frac{\Pr[C = c | M = m] \Pr[M = m]}{\Pr[C = c]} = \\ &= \frac{\frac{1}{2^l} \Pr[M = m]}{\frac{1}{2^l}} \\ &= \Pr[M = m]. \end{aligned} \tag{4}$$

□

However, from one-time padding, we can see the key length equal to the message length, which means in order to encryption l-bit message, l-bit key should be shared before. Actually this is the limitation of all perfectly secret encryption.

Theorem 2. *If* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a perfectly secret encryption scheme associated with message space* $\mathcal{M}$ *and key space* $\mathcal{K}$*, we have*

$$|\mathcal{K}| \geq |\mathcal{M}|.$$

Proof. We assume $|\mathcal{K}| < |\mathcal{M}|$, and show that Π can not be perfectly secret. For a ciphertext $c^* \in \mathcal{C}$, define

$$\mathcal{M}(c^*) := \{m \in \mathcal{M} \mid m = \mathsf{Dec}(K, c^*), \text{ for } K \in \mathcal{K}\}.$$

Since Dec is deterministic, $|M(c^*)| \leq |\mathcal{K}| < |M|$. There must exist a m' from $\mathcal{M}$ such that $m' \notin \mathcal{M}(c^*)$ (WOLG, assume the distribution of message is uniform). Thus,

$$0 = \Pr[M = m' \mid C = c^*] \neq \Pr[M = m'] = \frac{1}{2^l},$$

Π can not be perfectly secret. □

From this theorem, we can see the key length $\geq$ the message length is the inherent limitation of perfectly secret encryption.

3 Computational security

We could break the limitation of perfect security by considering security against computational bounded adversary, rather than unbounded adversary. This make sense since unbounded adversary does not exist in our real-life world. What we are facing is the probabilistic polynomial time (PPT) adversary.

Roughly, we say a scheme is computational secure, if any PPT adversary successfully breaks the scheme with a “small" probability. But what kind of probability could be taken as small enough.

Definition 2 (Negligible function)**.** *A positive function* $f : \mathbb{N} \to \mathbb{R}$ *is negligible if for every positive polynomial* $p(\cdot)$ *there exists an integer* N_p *such that for all integer* $x > N_p$ *(or if for every positive polynomial* $p(\cdot)$ *and all sufficiently large* x*),* $f(x) < \frac{1}{p(x)}$*. We generally denote an arbitrary negligible function by* negl*.*

Lemma 1. *Prove the following properties*

1. *if* negl_1 *and* negl_2 *are negligible functions,* $\mathsf{negl}_1 + \mathsf{negl}_2$ *is also a negligible function.*
2. *Assume* negl *is negligible, so does* $p(\cdot) \cdot \mathsf{negl}$ *for any positive polynomial* $p(\cdot)$*.*

Proof. For every polynomial $p(\cdot)$, $\exists\ N_1, N_2$ s.t.

$$\mathsf{negl}_1(n) < \frac{1}{2p(n)},\ \forall\ n > N_1,$$

$$\mathsf{negl}_2(n) < \frac{1}{2p(n)},\ \forall\ n > N_2.$$

Define $N \stackrel{def}{=} max\{N_1, N_2\}$. Then we have $\forall\ n > N$

$$\mathsf{negl}_1(n) + \mathsf{negl}_2(n) < \frac{1}{2p(n)} + \frac{1}{2p(n)} = \frac{1}{p(n)}.$$

For every positive polynomial $q(\cdot)$, $\exists\ N$ s.t. $\forall\ n > N$

$$\mathsf{negl}(n) < \frac{1}{p(n)q(n)},\ \text{then, } p(n) \cdot \mathsf{negl}(n) < p(n) \cdot \frac{1}{p(n)q(n)} = \frac{1}{q(n)}.$$

□

Remark 3. *$\frac{1}{2^n}$ is obviously a negligible function, while $\frac{1}{n^{1000}}$ is not.*

Definition 3. *An encryption scheme Π is said to be computational secure if for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ successfully breaks scheme Π is a negligible function of input length (or the probability is negligible).*

4 IND-eavesdropper security and construction

We do not define what is "successfully breaks" in definition 3. Here, we give more details about this by introducing "indistinguishably (IND) -eavesdropper security". Roughly, an adversary successfully breaks the IND-eavesdropper security of a scheme if it can distinguish which of the two messages (chosen by itself) is encrypted in the ciphertext. Formally it is defined by the following experiment of scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ between the adversary and challenger.

$\mathsf{Exp}^{ind-eva}_{\Pi}(\mathcal{A})$

1. The challenger chooses $b \leftarrow \{0, 1\}$ to indicate which message is encrypted
2. The challenger generates $(K, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3. Adversary $\mathcal{A}$ chooses and sends (M_0, M_1) to the challenger
4. The challenger runs $C^* = \mathsf{Enc}(K, M_b)$ and returns back C^*
5. $\mathcal{A}(C^*)$ returns b' as the guess of b
6. Return 1 if $b = b'$, else 0.

Definition 4 (IND-eav Security)**.** *The IND-eav-advantage of an adversary against IND-eavesdropper security of Π is defined as*

$$Adv^{ind-eav}_{\P 1}(\mathcal{A}) := |\Pr[\mathit{Exp}^{ind-eva}_{\Pi}(\mathcal{A}) \Rightarrow 1] - 1/2|.$$

Π is said to be IND-eva secure if for any PPT adversary, IND-eav-advantage is a negligible function of λ.

4.1 Construction from pseudorandom generator

We first introduce the primitive of pseudorandom generator, then construct IND-eva secure encryption from pseudorandom generator.

Definition 5 (Pseudorandom generator (PRG))**.** *Let G be a deterministic polynomial time algorithm takes $s \in \{0,1\}^n$ as input and outputs $G(s)$ of length $\ell(n)$. We say G is a pseudorandom generator (PRG), if it satisfies*

- $\ell(n) > n$
- *for any PPT algorithm $\mathcal{A}$, there exists a negligible function negl such that*

$$\Pr[\mathcal{A}(G(s)) = 1 \mid s \in \{0,1\}^n] - \Pr[\mathcal{A}(r) = 1 \mid r \in \{0,1\}^{\ell(n)}] \leq \mathsf{negl},$$

where the first probability is taken over the randomness of $\mathcal{A}$ and randomness of s, and the second one is taken over the randomness of $\mathcal{A}$ and the randomness of r.

We leave the construction of PRG in the next lecture. Here we assume the existence of PRG and based on it construct encryption scheme. Assume G with output length $\ell(n)$ is a secure PRG. A scheme $\Pi_1 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with fix length, introduced in the following, achieves the IND-eav security. Fix an integer $\ell(n)$ and set $\mathcal{M} = \mathcal{C} = \{0,1\}^\ell$, and $\mathcal{K} = \{0,1\}^n$.

KeyGen (1^n) On input 1^n, generate $K \leftarrow \{0,1\}^n$ and output (K, K) as secret keys.

$\mathsf{Enc}(K, M)$ On input K and message M, generate $C = G(K) \oplus M$ as the ciphertext.

$\mathsf{Dec}(K, C)$ On input K and the ciphertext C, return $M = G(K) \oplus C$ as the message.

The correctness is trivial.

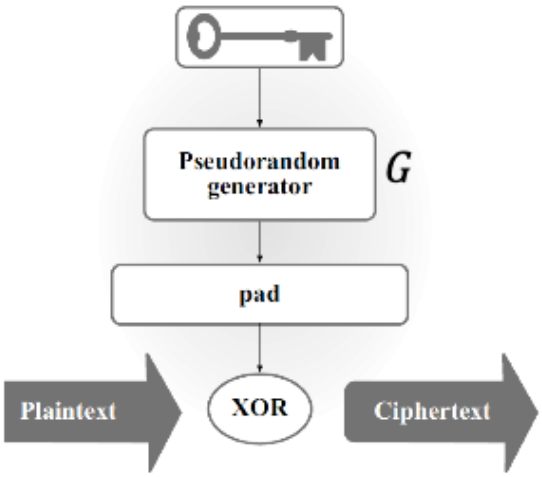

Figure 1: Illustration of Π_1 according to Fig 3.2 of [KL20]

Theorem 3. *Under the assumption G is a secure PRG, $\Pi_1 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-eva secure, i.e, for any PPT adversary $\mathcal{A}$, the IND-eav advantage is negligible.*

Please refer to section 3.3.3 of [KL20] for the proof.

5 IND-CPA security and construction

IND-eavesdropper is a very weak security aim. Actually, the adversary can do more things than just receives the ciphertext and guess which message is encrypted in the aim ciphertext. It can choose a message and asked the challenger (or user) to generate the ciphtext which may help it to attack the aim ciphertext. For example, in World War II, British placed naval mines at certain locations, knowing that the Germans—when finding those mines—would encrypt the locations and send them back to Germany. Thus, it is necessary to define a stronger security.

We abstract adversary's capability of choosing a message and getting the corresponding ciphertext by allowing adversary $\mathcal{A}$ to ask the algorithm $\mathsf{Enc}(K, \cdot)$ with any message it wants. We say this as allowing $\mathcal{A}$ to query the encryption oracle $\mathsf{Enc}(K, \cdot)$ with message m and receive the ciphertext $\mathsf{Enc}(K, m)$. This is usually denoted as $\mathcal{A}^{\mathsf{Enc}(K,\cdot)}$.

The resulting security is indistinguishably chosen plaintext security and defined via the following experiment. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme.

$\mathsf{Exp}^{ind-cpa}_{\Pi}(\mathcal{A})$

1. The challenger chooses $b \leftarrow \{0, 1\}$
2. The challenger generates $(K, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3. $(M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{Enc}(K,\cdot)}$ // here, $\mathcal{A}^{\mathsf{Enc}(K,\cdot)}$ means $\mathcal{A}$ can query the encryption oracle $\mathsf{Enc}(K, \cdot)$ with any message it wants
4. The challenger runs $C^* = \mathsf{Enc}(K, M_b)$ and returns back C^*
5. $\mathcal{A}^{\mathsf{Enc}(K,\cdot)}(C^*)$ returns b' as the guess of b
6. Return 1 if $b = b'$, else 0.

query $\mathsf{Enc}(K, \cdot)$ with m

1. return $\mathsf{Enc}(K, m)$

Definition 6 (IND-CPA Security)**.** *The IND-CPA-advantage of an adversary against IND-CPA security of* Π *is defined as*

$$Adv^{ind-cpa}_{\Pi}(\mathcal{A}) := |\Pr[Exp^{ind-cpa}_{\Pi}(\mathcal{A}) \Rightarrow 1] - 1/2|.$$

Π *is said to be IND-CPA secure if for any PPT adversary, IND-CPA-advantage is a negligible function of* λ*.*

5.1 Pseudorandom Function(PRF))

We first introduce the primitive of pseudorandom function, then construct IND-CPAsecure encryption from pseudorandom function.

Let $F : \{0,1\}^n \times \{0,1\}^{in} \to \{0,1\}^{out}$ be an efficient function (where *in* and *out* is a polynomial of n). For each $K \in \{0,1\}^n$, we get a function $F_K : \{0,1\}^{in} \to \{0,1\}^{out}$ defined by $F_K(X) = F(K, X)$. Let Func_n be the set of all the functions mapping $\{0,1\}^{in}$ to $\{0,1\}^{out}$. The size of Func_n is $2^{out2^{in}}$.

Definition 7 (Pseudorandom function (PRF))**.** *Let $F : \{0,1\}^n \times \{0,1\}^{in} \to \{0,1\}^{out}$ be an efficient function. We say F_K is a pseudorandom function (PRF), if for any PPT algorithm $\mathcal{A}$, there exists a negligible function negl of n such that*

$$\Pr[\mathcal{A}^{F_K(\cdot)}(1^n) = 1] - \Pr[\mathcal{A}^{f(\cdot)}(1^n) = 1] \leq \mathsf{negl},$$

where $\mathcal{A}^{F_K(\cdot)}$ (resp. $\mathcal{A}^{f(\cdot)}$) means $\mathcal{A}$ can query $F_K(\cdot)$ (resp. $f(\cdot)$) with any input it wants, f is a random function from Func_n, and the probability is taken over the randomness of $\mathcal{A}$.

We leave the construction of PRF in the next lecture. Here we assume the existence of PRF and based on it construct encryption scheme.

5.2 Construction

Assume F is a PRF. A scheme $\Pi_2 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, introduced in the following, achieves the IND-CPAsecurity with $\mathcal{M} = \{0,1\}^out$, $\mathcal{K} = \{0,1\}^n$, and $\mathcal{C} = \{0,1\}^{in} \times \{0,1\}^{out}$.

KeyGen (1^n) On input 1^n, generate $K \leftarrow \{0,1\}^n$ and output (K, K) as secret keys.

$\mathsf{Enc}(K, M)$ On input K and message M, choose randomness $r \leftarrow \{0,1\}^{in}$, compute $F_K(r) \oplus M$. Set the ciphertext as $C =< r, F_K(r) \oplus M >$

$\mathsf{Dec}(K, C)$ On input K and the ciphertext $C =< c_1, c_2 >$, return $M = F_K(c_1) \oplus c_2$ as the message.

The correctness is trivial.

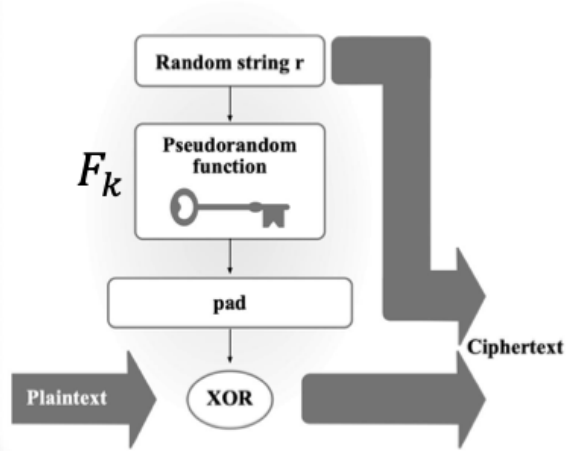

Figure 2: Illustration of Π_2 according to Fig 3.3 of [KL20]

Theorem 4. *Under the assumption F is a secure PRF, $\Pi_2 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-CPA secure, i.e, for any PPT adversary $\mathcal{A}$, the IND-eav advantage is negligible.*

Please refer to section 3.5.2 of [KL20] for the proof.

6 IND-CCA security and authenticated encryption

IND-CPA is still weak since the adversary can do more things than chosen plaintext. It may have the capability to choose a ciphertext and asked the challenger (or user) to find the corresponding plaintext. For example, in the case of CAPTCHA [Wik], the adversary (the client of CAPTCHA) can generate any ciphertext and receives the plaintext fron CAPTCHA server. Thus, it is necessary to define a stronger security.

We abstract adversary's capability of choosing a ciphertext and getting the corresponding plaintext by allowing adversary $\mathcal{A}$ to ask the algorithm $\mathsf{Dec}(K, \cdot)$ with any ciphertext (expect the aim ciphertext) it wants. We say this as allowing $\mathcal{A}$ to query the decrption oracle $\mathsf{Dec}(K, \cdot)$ with ciphertext $C \neq C^*$ and receive $\mathsf{Dec}(K, C)$. This is usually denoted as $\mathcal{A}^{\mathsf{Dec}(K,\cdot)}$.

The resulting security is indistinguishably chosen ciphertext security (IND-CCA) and defined via the following experiment. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme.

$\mathsf{Exp}^{ind-cca}_{\Pi}(\mathcal{A})$

1. The challenger chooses $b \leftarrow \{0, 1\}$
2. The challenger generates $(K, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3. $(M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{Enc}(K,\cdot),\mathsf{Dec}(K,\cdot)}$ // here, $\mathcal{A}^{\mathsf{Enc}(K,\cdot),\mathsf{Dec}(K,\cdot)}$ means $\mathcal{A}$ can query the encryption oracle $\mathsf{Enc}(K, \cdot)$ with any message it wants, and query the decryption oracle $\mathsf{Dec}(K, \cdot)$ with any ciphertext it wants
4. The challenger runs $C^* = \mathsf{Enc}(K, M_b)$ and returns back C^*
5. $\mathcal{A}^{\mathsf{Enc}(K,\cdot),\mathsf{Dec}(K,\cdot)}(C^*)$ returns b' as the guess of b //$\mathcal{A}^{\mathsf{Enc}(K,\cdot),\mathsf{Dec}(K,\cdot)}$ means $\mathcal{A}$ can query the encryption oracle $\mathsf{Enc}(K, \cdot)$ with any message it wants, and query the decryption oracle $\mathsf{Dec}(K, \cdot)$ with any ciphertext (except C^*) it wants
6. Return 1 if $b = b'$, else 0.

query $\mathsf{Enc}(K, \cdot)$ with m

1. return $\mathsf{Enc}(K, m)$

Definition 8 (IND-CCA Security)**.** *The IND-CCA-advantage of an adversary against IND-CCA security of Π is defined as*

$$Adv^{ind-cpa}_{\Pi}(\mathcal{A}) := |\Pr[\mathsf{Exp}^{ind-cpa}_{\Pi}(\mathcal{A}) \Rightarrow 1] - 1/2|.$$

Π is said to be IND-CCA secure if for any PPT adversary, IND-CCA-advantage is a negligible function of λ.

We build IND-CCA secure encryption from an IND-CPA secure encryotion and massage authenticated code defined in the following.

6.1 Massage authenticated code

A massage authenticated code (MAC) scheme MAC consists of three public algorithms $(\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$.

$\mathsf{KeyGen}\ (1^\lambda)$ On input 1^λ, generate $K \leftarrow K$ and output a pair key (K, K) as secret keys.

$\mathsf{Mac}_K(M)$ This is a probabilistic or deterministic algorithm. On input secret key K and message M, generate and output a tag t.

$\mathsf{Ver}_K(M, t)$ This is a deterministic algorithm. On input secret key K a message M and a tag t, return 1 to indicate valid and 0 to indicate invalid.

Correctness requires that for all $(K, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathsf{Ver}_K(m, \mathsf{Mac}_K(M))$ holds for all $M \in \mathcal{M}$.

Security of MAC requires that any adversary can not forge the MAC given several message and MAC pairs. The formal security of $\mathsf{MAC} = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$ against strong existentially unforgeable under an adaptive chosen-message attack (sUF-CMA) is defined via the following experiment.

$\mathsf{Mac\text{-}forge}_{\mathsf{MAC}}(\mathcal{A})$

1. $(K, K) \leftarrow \mathsf{KeyGen}(1^n)$
2. On input 1^n, adversary $\mathcal{A}$ is given the oracle access to $\mathsf{Mac}_K(\cdot)$. Let $Query$ be the list of $\mathcal{A}$'s queries and corresponding answers.
3. $\mathcal{A}$ returns (m, t) and successes if $\mathsf{Ver}_K(t, t) = 1$ and $(m, t) \notin Query$. If $\mathcal{A}$ successes, return 1, other wise return 0.

Definition 9 (sUF-CMA)**.** *A MAC scheme* $\mathsf{MAC} = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$ *is said to be existentiall y unfor g eable under an ada p tive chosen-message attack (sUF-CMA) secure, if for any PPT adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{Mac\text{-}forge}_{\mathsf{MAC}}(\mathcal{A}) \to 1] \leq \mathsf{negl}$$

Given a PRF F, the following $\mathsf{MAC} = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$ is a sUF-CMA secure MAC with fixed-length message.

$\mathsf{KeyGen}\ (1^\lambda)$ On input 1^λ, generate $K \leftarrow K$ and output a pair key (K, K) as secret keys.

$\mathsf{Mac}_K(M)$ On input K and message M, compute $t = F_K(M)$.

$\mathsf{Ver}_K(M,t)$ On input K a message M and a tag t, return 1 if $t = F_K(M)$, otherwise 0.

There are several ways to extend the domain of MAC. At the end of Section 7, HMAC with arbitrary input length is given.

6.2 IND-CPA and MAC ⇒ IND-CCA

Assume $\Pi_2 = (\cdot, \mathsf{Enc}, \mathsf{Dec})$ is an IND-CPA secure encryption and $\mathsf{MAC} = (\cdot, \mathsf{Mac}, \mathsf{Ver})$ is a sUF-CMA secure MAC. An IND-CCA secure encryption $\Pi_3 = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ follows.

KeyGen' (1^n) On input 1^n, generate $K_1, K_2 \leftarrow \{0,1\}^n$ and output $(K_1||K_2, K_1||K_2)$ as secret keys.

$\mathsf{Enc}'(K_1||K_2, M)$ On input $K = K_1||K_2$ and message M, compute $c_1 = \mathsf{Enc}(K_1, M)$. Compute $c_2 = \mathsf{Mac}_{K_2}(c_1)$. Set the ciphertext as $C =< c_1, c_2 >$

$\mathsf{Dec}'(K, C)$ On input $K = K_1||K_2$ and the ciphertext $C =< c_1, c_2 >$, if $\mathsf{Ver}_{K_2}(c_1, c_2) = 0$ abort. Return $\mathsf{Dec}(K_1, c_1)$ as the message.

Correctness of this scheme is guaranteed by the correctness of Π_2 and MAC.

Theorem 5. *Under the assumption that Π_2 is IND-CPA secure and MAC is sUF-CMA secure, Π_3 is IND-CCA secure.*

Please refer to [KL20, Theorem 4.19] for the proof.

7 Hash function

Hash functions are functions that take inputs of some length and compress them into fixed-length outputs. Let n be the fixed length. Hash function is defined as,

$$H : \{0,1\}^* \to \{0,1\}^n. \tag{5}$$

Generally, $n = 128, 160, 192$, or 256.

Concrete hash functions includes MD5 [Riv92], SHA1 [RO05], SHA2 [GD95], and SHA3 [D+15]. Currently, MD5 and SHA-1 are insecure due to practical attacks of [WY05], [SBK+17].

Two important properties of hash function are collision resistant and one-wayness.

For an adversary $\mathcal{A}$, and security parameter λ, we define collision finding experiment $\mathsf{Exp}^{cr}_H(\mathcal{A})(\lambda)$ and one-wayness experiment respectively.

$\mathsf{Exp}^{cr}_H(\mathcal{A})(\lambda)$

1. $X_1, X_2 \leftarrow \mathcal{A}(H, \lambda)$
2. IF $X_1 \neq X_2$ and $H(X_1) = H(X_2)$, return 1
3. ELSE, return 0.

$\mathsf{Exp}_H^{ow}(\mathcal{A})(\lambda)$

1. $X \leftarrow \{0,1\}^*$, $Y = H(M)$
2. $X' \leftarrow \mathcal{A}(H, Y, \lambda)$
3. IF $H(X') = Y$, return 1
4. ELSE, return 0.

Definition 10 (Collision resistance)**.** *A hash function is said to be collision resistant if for any PPT adversary $\mathcal{A}$, there is a negligible function* negl *such that*

$$\Pr[\mathit{Exp}_H^{cr}(\mathcal{A})(\lambda) \Rightarrow 1] = \mathsf{negl}.$$

Definition 11 (One-wayness)**.** *A hash function is said to be one way if for any PPT adversary $\mathcal{A}$, there is a negligible function* negl *such that*

$$\Pr[\mathit{Exp}_H^{cr}(\mathcal{A})(\lambda) \Rightarrow 1] = \mathsf{negl}.$$

Theorem 6. *If a hash function is collision resistant, then it is one way.*

Proof. We assume that there exists a PPT adversary $\mathcal{A}$ to break the one-wayness, then there exists an efficient adversary $\mathcal{B}$ to break the collision resistance (by querying $\mathcal{A}$). $\mathcal{B}$ works as following.

$\mathcal{B}^{\mathcal{A}}(H, \lambda)$

1. Pick $X \leftarrow \{0,1\}^*$, and give $H, Y = H(M), \lambda$ to $\mathcal{A}$
2. On receiving X' from $\mathcal{A}(H, Y, \lambda)$
3. Output (X, X')

Since $|\{0,1\}^*| \gg n$, the probability $X = X'$ is negligible. On the condition $X \neq X'$, if the probability that $\mathcal{A}$ breaks one-wayness is ε, then the probability that $\mathcal{B}$ breaks collision resistance is ε. □

Theorem 7. *If a hash function is one-way it is not necessary to be collision resistant.*

Proof. Suppose $H : \{0,1\}^* \to \{0,1\}^n$ is one-way, the following function

$$\tilde{H}(x) = \begin{cases} 0^n, & \text{if } x = 0 \text{ or } 1. \\ H(x), & \text{otherwise.} \end{cases}$$

is also one-way. However, it is not collision-resistant, since inputs 1 and 0 has the same image.

□

For general Birthday attack on hash functions, please refer to [KL20, Sec. 5.4.1] for more details.

HMAC By combining collision resistant hash function with MAC with fixed length, using the Hash-then-MAC paradigm, we can construct a MAC for arbitrary length. Roughly, assume $\mathsf{MAC} = (\mathsf{KeyGen}, \mathsf{Mac}, \mathsf{Ver})$ is a MAC with fixed length, the mew MAC run $\mathsf{Mac}'(K, M) = \mathsf{MAC}(K, H(M))$

References

[D+15] Morris J Dworkin et al. Sha-3 standard: Permutation-based hash and extendable-output functions. 2015.

[GD95] Patrick Gallagher and Acting Director. Secure hash standard (shs). *FIPS PUB*, 180:183, 1995.

[KL20] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[Riv92] Ronald Rivest. The md5 message-digest algorithm. Technical report, 1992.

[RO05] Vincent Rijmen and Elisabeth Oswald. Update on sha-1. In *Cryptographers' Track at the RSA Conference*, pages 58–71. Springer, 2005.

[SBK+17] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full sha-1. In *Annual international cryptology conference*, pages 570–596. Springer, 2017.

[Wik] Wikipedia. `https://en.wikipedia.org/wiki/CAPTCHA`.

[WY05] Xiaoyun Wang and Hongbo Yu. How to break md5 and other hash functions. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 19–35. Springer, 2005.