CS252: Cryptography

Maple Hu

Update: August 11, 2023

Contents

1 Preliminary **3**

1.1 Elementary Number Theory 3

1.2 Abstract Algebra 4

2 Introduction **6**

2.1 Definition of Cryptography 6

2.2 Classic Cryptography 6

3 Private-key Encryption Scheme **8**

3.1 Perfect Secrecy 8

3.2 Perfect Indistinguishability 9

3.3 IND-EAV 11

3.4 Computationally Indistinguishable 14

3.5 Pseudorandomness 14

3.6 IND-m-EAV 17

3.7 IND-CPA 18

3.8 Pseudorandom Function 20

3.9 Pseudorandom Permutation 23

3.10 Models of Operation 24

3.11 Message Authentication Code 29

3.12 Chosen-Cipher-Text Attack 34

3.13 Hash Function 35
3.14 Data Encryption Standard (DES) 40
3.15 Advanced Encryption Standard 44
3.16 Discrete Logarithm 46
3.17 Key Exchange 47

4 Public-Key Encryption **50**

4.1 Introduction 50
4.2 EIGamal Encryption 51
4.3 RSA 52
4.4 Message Authentication Code 53

1 Preliminary

1.1 Elementary Number Theory

Definition 1.1 Let $\mathbb{Z}$ be the set of integers. Let $a, b \in \mathbb{Z}$ and $a \neq 0$. If exists a $c \in \mathbb{Z}$ such that $b = ac$, then we say a divides b, simply $a|b$.

Definition 1.2 If a integers is prime if the only positive divisors of n are 1 and n.

Theorem 1.1 *(Fundamental Theorem of Arithmetic) Every integer $n > 1$ can be uniquely written as $n = p_1^{e_1} \cdots p_r^{e_r}$, where $p_1, \cdots, p_r$ are distinct primes, $p_1 < \cdots < p_r$ and $e_1, \cdots, e_r \geq 1$.*

Theorem 1.2 *(Division Algorithm) Let α, β be integers and $b > 0$. Then there are unique integers q, r such that*

$$a = bq + r \text{ and } 0 \leq r < b$$

Definition 1.3 Let non-zero $a, b \in \mathbb{Z}$. The common divisor d of a and b is such that $d|a$ and $d|b$. The greatest common divisor of a and b is called greatest common divisor, simply $\gcd(a, b)$.

Theorem 1.3 *Given non-zero $a, b, c \in \mathbb{Z}$.*

- *If $c|ab$ and $\gcd(a, c) = 1$, then $c|b$.*
- *If p is a prime and $p|ab$, then $p|a$ or $p|b$.*
- *Let N be non-zero integers. If $a|N$, $b|N$ and $\gcd(a, b) = 1$, then $ab|N$.*

Definition 1.4 Let $n \in \mathbb{Z}^+$ and $a, b \in \mathbb{Z}$. If $n|(a - b)$, we write $a \equiv b \mod n$.

Theorem 1.4 *Let $n \in \mathbb{Z}^+$. For any $a \in \mathbb{Z}$, there is a unique integer r such that $0 \leq r < n$ and $a \equiv r \mod n$.*

Definition 1.5 Let $n \in \mathbb{Z}^+$ and $a \in \mathbb{Z}$. The $[a]_n = a + n\mathbb{Z} = \{a + nx : x \in \mathbb{Z}\}$ is called the residue class of $a \mod n$.

Definition 1.6 Let $n > 0$ be an integer. The $\mathbb{Z}_n = \{[0]_n, [1]_n, \cdots, [n-1]_n\}$ is defined as the set of all reside classes modulo n.

Definition 1.7 Let $n \in \mathbb{Z}^+$ and $\mathbb{Z}_n = \{[0]_n, [1]_n, \cdots, [n-1]_n\}$. Define

- addition: $[a]_n + [b]_n = [a + b]_n$.
- subtraction: $[a]_n - [b]_n = [a - b]_n$.
- multiplication: $[a]_n \cdot [b]_n = [a \cdot b]_n$

The equalities in definition 1.7 should be proved. For example, we define $[a]_n \cdot [b]_n = \{ x_1 + x_2 \mid x_1 \in [a]_n \wedge x_2 \in [b]_n \}$ (that is the addition of sets), then you should prove the set $[a]_n \cdot [b]_n$ is equal to $[a \cdot b]_n$. Let us do this proof, suppose $x_1 = a + x_1' n$ and $x_2 = b + x_2' n$. We can get

$$x_1 \cdot x_2 = ab + (ax_2' + bx_1')n + x_1' x_2' n^2 \equiv ab \mod n.$$

Thus $[a]_n \cdot [b]_n \subseteq [a \cdot b]_b$. For any $ab + x_3' n \in [a \cdot b]_n$, it could be decomposed to

$$ab + x_3' n = (a + x_1' n)(b + x_2' n) - (ax_2' + bx_1' - x_3')n - x_1' x_2' n^2 \equiv (a + x_1' n)(b + x_2' n) \mod n.$$

Thus $[a \cdot b]_b \subseteq [a]_n \cdot [b]_n$. Therefore $[a \cdot b]_n = [a]_n \cdot [b]_n$.

Definition 1.8 Let $n \in \mathbb{Z}^+$ and $[a]_n \in \mathbb{Z}_n$. A $[s]_n \in \mathbb{Z}_n$ is called an inverse of $[a]_n$ if $[a]_n[s]_n = [1]_n$. If $[a]_n[s]_n = [1]_n$, define $\frac{[b]_n}{[a]_n} = [b]_n \cdot [s]_n$ (both multiply $[b]_n$).

Theorem 1.5 *Let $n \in \mathbb{Z}^+$, then $[a]_n \in \mathbb{Z}_n$ have an inverse if and only if* $\gcd(a, n) = 1$.

Proof. By $[a]_n[s]_n = [as]_n$. Suppose $[as]_n = [1]_n$, then there is a x such that $as = 1 + nx$. Let c be common divisor of a and n. By $c|as$ and $as = 1 + nx$, we can get $c|1$. Thus $c = 1$, so $\gcd(a, n) = 1$.

Suppose $\gcd(a, n) = 1$. Then we have $as + nt = 1$, that is $as = 1 - nt$. Thus $[as]_n = [1]_n$.

Definition 1.9 Let $n \in \mathbb{Z}^+$. Define $\mathbb{Z}_n^* = \{\, [a]_n \in \mathbb{Z}_n \mid \gcd(a, n) = 1 \,\}$.

Example 1.1 If p is prime, then $\mathbb{Z}_p^* = \{[1], [2], \cdots, [p-1]\}$.

1.2 Abstract Algebra

Definition 1.10 A group $(G, \bullet)$ consists of a set G and binary operator $\bullet$ satisfied

- $\forall a, b \in G$, $a \bullet b \in G$.
- $\forall a, b, c \in G$, $a \bullet (b \bullet c) = (a \bullet b) \bullet c$.
- $\exists e \in G, \forall a \in G$, $a \bullet e = e \bullet a = a$.
- $\forall a \in G, \exists b \in G$ such that $a \bullet b = b \bullet a = e$.

Definition 1.11 A group $(G, \bullet)$ is called an Abelian group if

$$\forall a, b \in G, a \bullet b = b \bullet a.$$

Lemma 1.6 *$(\mathbb{Z}_n, +)$ and $(\mathbb{Z}_n^*, \cdot)$ are Abelian group.*

Definition 1.12 A finite field is a finite set $\mathbb{F}$ along with two binary operator $+, \cdot$ satisfied

- $\mathbb{F}$ is an Abelian group with respect to the operator $+$.
- $\mathbb{F}\backslash\{0\}$ is an Abelian group with respect to the operator $\cdot$.
- Distributivity: For all $a, b, c \in \mathbb{F}$, $a \cdot (b + c) = ab + ac$.

Example 1.2 Let p be a prime, then $(\mathbb{Z}_p, +, \cdot)$ is a finite field.

Remark A finite field always has p^k elements for a prime p and an integer $k > 0$. Conversely, for any prime p and any integer $k > 0$, there is a finite field of p^k elements.

Example 1.3 Define $\mathbb{Z}_2[X] = \{\, a_0 + a_1 X + \cdots + a_d X^d \mid d = 0, 1, \cdots \text{ and } a_1 \in \mathbb{Z}_2 \,\}$. The $(\mathbb{Z}_2[X], +, \cdot)$ is field such that

- $\sum_{i=0}^{d} a_i X^i + \sum_{i=0}^{d} b_i X^i = \sum_{i=0}^{d} (a_i \oplus b_i) X^i$.
- $\sum_{i=0}^{d} a_i X^i \cdot \sum_{i=0}^{d} b_i X^i = \sum_{i=0}^{2d} (\sum_{\oplus j=0}^{i} a_j \cdot b_{i-j}) X^i$.

where $a_i \oplus b_i$ satisfies

$$a_i \oplus b_i \begin{cases} 0 & a_i = 0, b_i = 0, \\ 1 & a_i = 1, b_i = 0 \text{ or } a_i = 0, b_i = 1, , \\ 0 & a_i = 1, b_i = 1 \end{cases}$$

$\sum_{\oplus}$ is sum with $\oplus$.

Example 1.4 Let $\mathbb{F}_{2^2} = \mathbb{Z}_2[X]/(X^2+X+1) = \{0, 1, X, 1+X\}$. The $(\mathbb{F}_{2^2}, +, \cdot)$ is a finite field such that

- $(a_0 + a_1X) + (b_0 + b_1X) = (a_0 \oplus b_0) + (a_1 \oplus b_1)X$.
- $(a_0 + a_1X) \cdot (b_0 + b_1X) = (a_0b_0 \oplus a_1b_1) + (a_0b_1 \oplus a_1b_0 \oplus a_1b_1)X$.

Note $X^2 + X + 1$ can be replaced with 0.

Definition 1.13 The order of a group G is the cardinality of G as a set.

Definition 1.14 When $|G| < \infty$, for every $a \in G$, the order of a is defined as the least integer $l > 0$ such that $a^l = 1$ ($la = 0$ for additive group).

Example 1.5 Given $\mathbb{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$, then we have $o(1) = 1, o(2) = 3, o(3) = 6, o(4) = 3, o(5) = 6, o(6) = 2$.

Theorem 1.7 *Let G be a multiplicative Abelian group of order m. Then for any $a \in G$, $a^m = 1$.*

Proof. Suppose $G = a_1, a_2, \cdots, a_m$. We have $aa_i \neq aa_j$ for any distinct $a_i, a_j \in G$. Since we can multiply a^{-1} to two side, that is $a_i \neq a_j$. Then we consider $aa_1 \cdot aa_2 \cdots aa_m$, that is actually the multiple of m distinct elements in G. Thus we have $aa_1 \cdot aa_2 \cdots aa_m = a_1a_2 \cdots a_m$, then we can get $a^m = 1$ by multiplying $a_1^{-1} \cdots a_m^{-1}$.

Theorem 1.8 *(Euler's Theorem) Let $n > 1$ and $\alpha \in \mathbb{Z}_n^*$. Then $\alpha^{\phi(n)} = 1$, where $\phi(n) = |\mathbb{Z}_n^*|$.*

Theorem 1.9 *(Fermat's Little Theorem) If p is a prime and $\alpha \in \mathbb{Z}_p$. Then $\alpha^p = \alpha$.*

Corollary *If p is a prime and $\alpha \in \mathbb{Z}_p$. Then $a^{-1} = a^{p-2}$.*

Definition 1.15 Let $(G, \bullet)$ be an Abelian group. A subset $H \subseteq G$ is called a subgroup of G if $(H, \bullet)$ is also a group, written $H \leq G$.

Example 1.6 $H = \{1\}$ is subgroup of $\mathbb{Z}_6^*$; $H = \{0, 2, 4\}$ is subgroup of $\mathbb{Z}_6^+$.

Theorem 1.10 *Let $(G, \bullet)$ be an Abelian group. Let $\langle g \rangle = \{g^k : k \in \mathbb{Z}\}$ be a subset of G, where $g \in G$. Then $\langle g \rangle \leq G$.*

Definition 1.16 Let $(G, \bullet)$ be an Abelian group. G is said to be cyclic if there exists $g \in G$ such that $G = \langle g \rangle$, g is called a generator of G.

Remark Let G be a finite group and let $g \in G$. Then $\langle g \rangle$ can be computed as $\{g^1, g^2, \cdots\}$.

2 Introduction

2.1 Definition of Cryptography

From "Introduction to Modern Cryptography", Katz, Lindell, 2015.

Definition 2.1 The study of mathematical techniques for securing digital information, systems, and distributed computations against ad

From "Handbook of Applied Cryptography", Menezes, van Oorschot, Vanstone, 1996.

Definition 2.2 The study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication.

From CNSSI-4009.

Definition 2.3 The discipline that embodies the principles, means, and methods for the providing information security, including confidentiality, data integrity, non-repudiation, and authenticity.

Remark Information reduces uncertainty and therefore reduces entropy. Shannon.

Remark (Kerckhoff's Principle) The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.

Kerckhoff's principle is actually means

- Security should solely rely on the secrecy of the key.
- Cryptographic algorithms can be made public.

2.2 Classic Cryptography

Definition 2.4 A private-ket encryption is PrivKE $= (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$, where Gen is key generation, Enc is encryption, Dec is decryption and $\mathcal{M}$ is plaintext space. Let $\mathcal{K}$ be secret key space and $\mathcal{C}$ is ciphertext space. For any $k \in \mathcal{K}$ and $m \in \mathcal{M}$, PrivKE satisfy

$$\text{Dec}(k, \text{Enc}(k, m)) = m$$

Definition 2.5 (Shift Cipher) Let $\mathcal{M} = \{a, b, \cdots, z\}^*, \mathcal{C} = \{A, B, \cdots, Z\}^*; \mathcal{K} = \{0, 1, \cdots, 25\}$. Define a encryption as follows.

- Gen: let $k \to \{0, 1, \cdots, 25\}$ and output k.
- $\text{Enc}(k, m_1 m_2 \cdots m_n)$: let $c_i = (m_i + k) \mod 26$ and output $c_1 c_2 \cdots c_n$.
- $\text{Dec}(k, c_1 c_2 \cdots c_n)$: let $m_i = (c_i - k) \mod 26$ and output $m_1 m_2 \cdots m_n$.

Shift cipher is not secure, we use brute force attack to try all possible secret keys and find the one used for encryption.

Definition 2.6 (Substitution Cipher) Let $\mathcal{M} = \{a, b, \cdots, z\}^*, \mathcal{C} = \{A, B, \cdots, Z\}^*$, and $\mathcal{K} = \{\, \sigma : \{a, b, \cdots, z\} \to \{A, B, \cdots, Z\} \mid \sigma \text{ is a bijection} \,\}$. Define a encryption as follows.

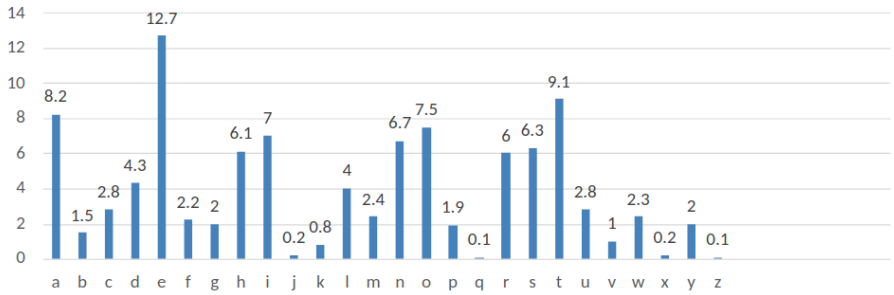


Figure 1: letter frequency

- Gen: let $\sigma \to \mathcal{K}$ and output σ.
- Enc$(k, m_1 m_2 \cdots m_n)$: let $c_i = \sigma(m_i)$ and output $c_1 c_2 \cdots c_n$.
- Dec$(k, c_1 c_2 \cdots c_n)$: let $m_i = \sigma^{-1}(c_i)$ and output $m_1 m_2 \cdots m_n$.

Though key space $|K| = 26!$ is large for brute-force attack but still not secure. Since every letter $m \in \{a, b, \cdots, z\}$ is mapped to a fixed letter, this is the frequencies of m and $\sigma(m)$ are equal for every m. The frequencies of individual letters are known in a normal English text.

Definition 2.7 (Vigenère Cipher) Let $\mathcal{K} = \mathcal{M} = \{a, b, \cdots, z\}^*$ and $\mathcal{C} = \{A, B, \cdots, Z\}^*$.

- Gen: let $k \to a, b, \cdots, z^*$, say $k = k_1 k_2 \cdots k_t$ and output k.
- Enc$(k, m_1 m_2 \cdots m_n)$: let $\lceil n/t \rceil = d$.
  - let $m = m_1 \cdots m_t || m_{t+1} \cdots m_{2t} || m_{(d-1)t+1} \cdots m_n$.
  - for each $c_i = m_i + k_{(i \mod t)+1}$.
  - output $c = c_1 c_2 \cdots c_n$.
- Dec$(k, c_1 c_2 \cdots c_n))$:

If the key length t of the cipher is known, it suffices to break t shift cipher by considering the following t sequences:

$$c_1 c_{1+t} c_{1+2t} \cdots$$
$$c_2 c_{2+t} c_{2+2t} \cdots$$
$$\vdots$$
$$c_t c_{2t} c_{3t} \cdots$$

which are backed to the shift cipher. Kasiski's method use the same segments such that their distance is a multiple of the key length t. It follows three steps.

- Search the ciphertext for pairs of identical segments of length at least three.
- Record the distances between the starting positions of any two segments.
- Output the greatest common divisor of there distances.

Another method for calculate k is index of coincidence algorithm

$$\sum_{i=1}^{26} \frac{n_i}{N} \cdot \frac{n_i - 1}{(N-1)} = 0.065$$

where n_i is the number of times the ith letter appears in the text and N is the total characters number in the text.

3 Private-key Encryption Scheme

3.1 Perfect Secrecy

Definition 3.1 $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ is perfectly secret if for any distribution $\mathcal{M}$, any $m \in \mathcal{M}$, and any $c \in \mathcal{C}$ with $\Pr[M = m] > 0$

$$\Pr[M = m | C = c] = \Pr[M = m]$$

Theorem 3.1 *$\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ is perfectly secret if and only if $\forall m, m' \in \mathcal{M}, \forall c \in \mathcal{C}$,*

$$\Pr[\text{Enc}(k, m) = c] = \Pr[\text{Enc}(k, m') = c] \text{ or } \Pr[C = c | M = m] = \Pr[C = c | M = m']$$

Proof. Firstly, we have

$$\Pr[\text{Enc}(K, m) = c] = \Pr[C = c | M = m]$$

for any $m \in \mathcal{M}$.

($\Rightarrow$). Suppose Π is perfectly secret satisfied $\Pr[M = m | C = c] = \Pr[M = m]$. We have

$$\begin{aligned}
\Pr[\text{Enc}(K, m) = c] &= \Pr[C = c | M = m] \\
&= \frac{\Pr[C = c \wedge M = m]}{\Pr[M = m]} \\
&= \frac{\Pr[M = m | C = c] \Pr[C = c]}{\Pr[M = m]} \\
&= \Pr[C = c]
\end{aligned}$$

($\Leftarrow$). Suppose $\forall m, m' \in \mathcal{M}, \forall c \in \mathcal{C}, \Pr[\text{Enc}(k, m) = c] = \Pr[\text{Enc}(k, m') = c]$. We have

$$\Pr[M = m | C = c] = \frac{\Pr[C = c | M = m] \, | \Pr[M = m]}{\Pr[C = c]}.$$

For the $\Pr[C = c]$, we have

$$\begin{aligned}
\Pr[C = c] &= \sum_{m' \in \mathcal{M}} \Pr[C = c | M = m'] \Pr[M = m'] \\
&= \sum_{m' \in \mathcal{M}} \Pr[C = c | M = m] \Pr[M = m'] \\
&= \Pr[C = c | M = m]
\end{aligned}$$

Thus we have $\Pr[M = m | C = c] = \Pr[M = m]$.

Definition 3.2 (One-time pad) Let $\mathcal{K} = \{0,1\}^n, \mathcal{M} = 0, 1^n, \mathcal{C} = 0, 1^n$.

- $\text{Gen}(1^n)$: let $k \leftarrow \{0,1\}^n$ and output k.
- $\text{Enc}(k, m)$: output $k \oplus c$.
- $\text{Dec}(k, c)$: output $k \oplus c$.

where $\oplus$ is the XOR operator.

Theorem 3.2 *One-time pad is perfectly secret.*

Proof. For any $m, m' \in \mathcal{M}$ and any $c \in \mathcal{C}$, we have

$\Pr[C = c|M = m] = \Pr[K \oplus m = c] = \Pr[K = c \oplus m] = 2^{-n}\Pr[C = c|M = m'] = \Pr[K \oplus m' = c] = \Pr[K = c \oplus m'] = 2^{-}$

Thus it is perfectly secret.

Note THe same secret key cannot be used more than once, if there are key ciphers $c = k \oplus m$ and $c' = k \oplus m'$, than we can get the information $m \oplus m' = c \oplus c'$. This information may break system security sometime.

Theorem 3.3 *If* $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ *is a perfectly secret encryption scheme with key space* $\mathcal{K}$*, then* $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof. Assume that $|\mathcal{K}| < |\mathcal{M}|$. For every $c \in \mathcal{C}$, we define

$$\mathcal{M}_c = \{m \in \mathcal{M} : \exists k \in \mathcal{K} \text{ such that } \text{Dec}(k, c) = m\}.$$

Then we have $|\mathcal{M}_c| \leq |\mathcal{K}| < |\mathcal{M}|$. There exists a $m \in \mathcal{M} \backslash \mathcal{M}_c$. Suppose M is the uniform distribution over $\mathcal{M}$. Then

$$\Pr[M = m] = \frac{1}{\mathcal{M}} > 0$$
$$\Pr[M = m|C = c] = 0$$

It is contradict to Π is perfectly secret.

3.2 Perfect Indistinguishability

Definition 3.3 Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ be a private-key encryption. Define an adversarial indistinguishability experiment $\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}$ in Figure 2.

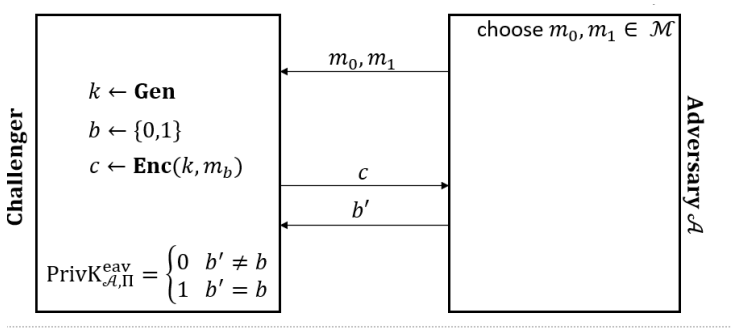


Figure 2: adversarial indistinguishability experiment

Definition 3.4 Π is perfectly indistinguishable if for every adversary $\mathcal{A}$ satisfied

$$\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1\right] = \frac{1}{2}.$$

Theorem 3.4 Π *is perfectly secret iff* Π *is perfectly indistinguishable.*

Proof. ($\Rightarrow$). We should show $\Pr[b = b'] = \frac{1}{2}$. That is

$$\begin{aligned}
\Pr[b = b'] &= \Pr[b' = 0|b = 0] + \Pr[b' = 1|b = 1] \\
&= \Pr[\mathcal{A}(m_0, m_1, \text{Enc}(k, m_0)) = m_0|b = 0] + \\
&\quad \Pr[\mathcal{A}(m_0, m_1, \text{Enc}(k, m_1)) = m_1|b = 1] \\
&= \Pr[\mathcal{A}(m_0, m_1, \text{Enc}(k, m_0)) = m_0] \Pr[b = 0] + \\
&\quad \Pr[\mathcal{A}(m_0, m_1, \text{Enc}(k, m_1)) = m_1] \Pr[b = 1]
\end{aligned}$$

where the last equality is based on the fact that $\Pr[\text{Enc}(k,m)=c] = \Pr[\text{Enc}(k,m')=c]$. Let $\text{Enc}(k,m_0)$ and $\text{Enc}(k,m_1)$ be two random variables, where for any fixed m_1, m_2 and k is uniform distribution. It is obvious that $\text{Enc}(k,m_0) \equiv \text{Enc}(k,m_1)$. Thus we have

$$\begin{aligned}\Pr[b=b'] &= \Pr[\mathcal{A}(m_0,m_1,\text{Enc}(k,m_0))=m_0]\frac{1}{2}+ \\ &\quad \Pr[\mathcal{A}(m_0,m_1,\text{Enc}(k,m_0))=m_1]\frac{1}{2} \\ &= \frac{1}{2}\end{aligned}$$

($\Leftarrow$). Assume exists m_0, m_1 and c such that $\Pr[C=c|M=m_0] \neq \Pr[C=c|M=m_1]$. Without loss of general, we assume

$$\Pr[C=c|M=m_0] > \Pr[C=c|M=m_1],$$

For every $y \in \mathcal{C}$, we make a choice called $i_y \in \{0,1\}$ satisfied

$$\Pr\left[C=y|M=m_{i_y}\right] \geq \Pr\left[C=y|M=m_{1-i_y}\right].$$

Then we construct a adversary as follows.

- Send (m_0, m_1) to the challenger;
- If y is received from the challenger, outputs $b' = i_y$.

Now consider the $\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}=1\right]$:

$$\begin{aligned}\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}=1\right] &= \Pr[b'=b] \\ &= \sum_{y\in\mathcal{C}} \Pr[b'=b|C=y]\Pr[C=y] \\ &= \sum_{y\in\mathcal{C}} \Pr[i_y=b|C=y]\Pr[C=y]\end{aligned}$$

in where

$$\begin{aligned}\Pr[i_y=b|C=y] &= \frac{\Pr[C=y|b=i_y]\Pr[b=i_y]}{\Pr[C=y]} \\ &= \frac{\Pr[C=y|b=i_y]}{2\Pr[C=y]} \qquad \Pr[b=i_y]=\frac{1}{2} \\ &= \frac{\Pr\left[C=y|M=m_{i_y}\right]}{2\Pr[C=y]} \begin{cases} \geq \dfrac{\Pr[C=y|M=m_1]}{2\Pr[C=y]} & y\neq c \\ > \dfrac{\Pr[C=y|M=m_1]}{2\Pr[C=y]} & y=c \end{cases}\end{aligned}$$

Thus we have

$$\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}=1\right] > \sum_{y\in\mathcal{C}} \frac{\Pr[C=y|M=m_1]}{2\Pr[C=y]}\Pr[C=y] = \frac{1}{2}.$$

Example 3.1 Let $\mathcal{K} = \{a,b,\cdots,z\} \cup \{a,b,\cdots,z\}^2, \mathcal{M} = \{a,b,\cdots,z\}^*$ and $\mathcal{C} = \{A,B,\cdots,Z\}^*$. Define a encryption scheme as follows.

- Gen: the key length is 1 or 2 with equal probability
  - $\Pr[K=k] = \frac{1}{2\cdot 26}$ for every $k \in \{a,b,\cdots,z\}$.
  - $\Pr[K=k] = \frac{1}{2\cdot 26^2}$ for every $k \in \{a,b,\cdots,z\}^2$.
- The encryption and decryption scheme are same as Vigenére cipher.

This scheme is not perfectly indistinguishable. We can construct an adversary as follows.

- choose $m_0 = xx$ and $m_1 = xy$, and give them to challenger.
- learns $c = c_0c_1$ from the challenger.
- if $c_0 = c_1$, output $b' = 0$. Otherwise output $b' = 1$.

We know $\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1\right] = \Pr[b' = b] = \frac{1}{2}\Pr[b' = 0|b = 0] + \frac{1}{2}\Pr[b' = 1|b = 1]$. Consider two cases respectively:

- when $b = 0 \wedge b' = 0$, we should find all possible keys such that $c_1 = c_2$.
  - $|k| = 1$, that is 26 cases.
  - $|k = k_1k_2| = 2 \wedge k_1 = k_2$, that is 26 cases.
- when $b = 1 \wedge b' = 1$, we should find all possible keys such that $c_1 \neq c_2$. Instead of considering straight case, it is easier to consider $c_1 = c_2$ cases and exclude them.
  - $|k = k_1k_2| = 2 \wedge k_1 = a \wedge k_1 = a - 1$, that is 26 cases.

Then we have

$$\begin{aligned}\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi} = 1\right] &= \frac{1}{2}\Pr[b' = 0|b = 0] + \frac{1}{2}\Pr[b' = 1|b = 1] \\ &= \frac{1}{2}\left(\frac{1}{2}\cdot\frac{26}{26} + \frac{1}{2}\cdot\frac{26}{26^2}\right) + \frac{1}{2}\left(1 - \frac{1}{2}\cdot\frac{26}{26^2}\right) \\ &= \frac{3}{4} > \frac{1}{2}\end{aligned}$$

3.3 IND-EAV

Definition 3.5 A is a polynomial-time algorithm if there is a polynomial $p(\cdot)$ such that $\forall x \in 0, 1^*$, the running time of $A(x) \leq p(|x|)$.

Definition 3.6 A function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if for any polynomial function $p(\cdot) > 0$, there exists N such that $f(n) < \frac{1}{p(n)}$ for all $n > N$.

Example 3.2 There are some negligible functions and non-negligible functions.

- negligible: 2^{-n}, $2^{-\sqrt{n}}$ and $n^{-\log n}$.
- non-negligible: $\frac{1}{n^{10000}}$ and $n^{\frac{-\log\log n}{\log n}}$.

Theorem 3.5 *Let $f(n)$ and $g(n)$ be negligible and let $p(n)$ be a polynomial. Then $f(n) + g(n)$, $p(n) \cdot f(n)$ are both negligible.*

Definition 3.7 Π has indistinguishable encryption in the presence of an eavesdropper (IND-EAV) if for all PPT adversaries $\mathcal{A}$ there is a negligible function negl$(\cdot)$ such that

$$\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \text{negl}(n).$$

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment. There is equivalent definition:

$$|\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2}| \leq \text{negl}(n).$$

Definition 3.8 Give a experiment $\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n)$, we fixed the b, then we call the new experiment as $\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, b)$.

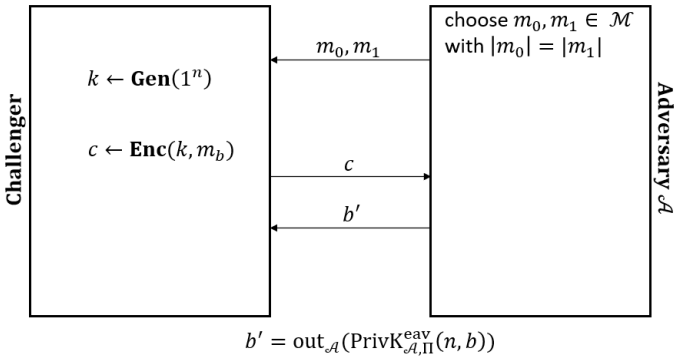


Figure 3: Experiment $\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, b)$

Definition 3.9 (Equivalent def) Let $b' = \text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, b))$. Π has indistinguishable encryption in the presence of an eavesdropper (IND-EAV) if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\text{negl}(\cdot)$ such that

$$|\Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 0)) = 1\right] - \Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 1)) = 1\right]| \leq \text{negl}(n).$$

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment.

Theorem 3.6 *Definition 3.7 is equivalent to Definition 3.9.*

Proof. We call Definition 3.7 IND-EVA1, and call Definition 3.9 IND-EVA2.

($\Rightarrow$) Suppose Π is IND-EVA1. Assume Π is not IND-EVA2. Then

$$\epsilon = |\Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 0)) = 1\right] - \Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 1)) = 1\right]|$$

is non-negligible for some PPT adversary $\mathcal{A}$. Without loss of generality, we assume

$$\Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 0)) = 1\right] - \Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 1)) = 1\right] \geq 0 \quad (1)$$

We show that Π is not IND-EAV1 (gives a contradiction). We construct a adversary $\mathcal{B}$ combined with $\mathcal{A}$ in Figure 4. The more subtle step is that adversary $\mathcal{B}$ output the contrary value of b'. Since (1) means the probability that $\mathcal{A}$

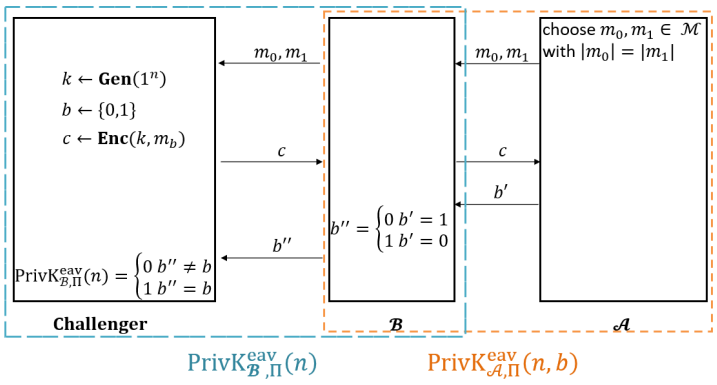


Figure 4: Combined adversary $\mathcal{A}$ and $\mathcal{B}$

guess 1 when $b = 0$ is large than the probability that $\mathcal{A}$ guess 1 when $b = 1$. Thus $\mathcal{B}$ output 0 when b' output 1. Similarly, we have

$$\Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 1)) = 0\right] = 1 - \Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 1)) = 1\right] \geq$$
$$\Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 0)) = 0\right] = 1 - \Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 0)) = 1\right]$$

that is $\mathcal{B}$ output 1 when $b' = 0$. Consider the $\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n)$, we have

$$\begin{aligned}
\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{B},\Pi}(n) = 1\right] &= \frac{1}{2}\Pr\left[b'' = 0 | b = 0\right] + \frac{1}{2}\Pr\left[b'' = 1 | b = 1\right] \\
&= \frac{1}{2}\Pr\left[b' = 1 | b = 0\right] + \frac{1}{2}\Pr\left[b' = 0 | b = 1\right] \\
&= \frac{1}{2}\Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 0)) = 1\right] + \frac{1}{2}(1 - \Pr\left[\text{out}_{\mathcal{A}}(\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n, 1)) = 1\right]) \\
&= \frac{1}{2} + \frac{\epsilon}{2}.
\end{aligned}$$

is non-negligible.

($\Leftarrow$) Suppose Π is IND-EAV2. Assume Π is not IND-EAV1. Then

$$\epsilon = |\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2}|$$

is non-negligible for some PPT adversary $\mathcal{A}$. Without loss of generality, we assume $\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2} \geq 0$. We show that Π is not IND-EAV2 (gives a contradiction). We could construct a adversary $\mathcal{B}$ combined with $\mathcal{A}$ in Figure 5. Adversary $\mathcal{B}$ take $\mathcal{A}$'s output as own output straightly. Then we have

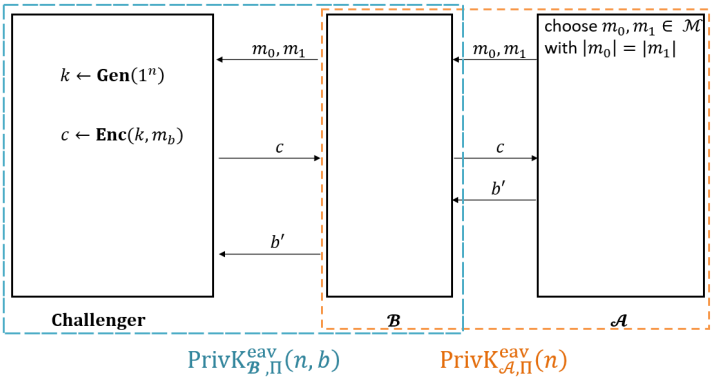


Figure 5: Combined adversary $\mathcal{A}$ and $\mathcal{B}$ (2)

$$\begin{aligned}
\Pr\left[\text{out}_{\mathcal{B}}(\text{PrivK}^{\text{eav}}_{\mathcal{B},\Pi}(n, 0)) = 1\right] &= \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 0 | b = 0\right] \\
\Pr\left[\text{out}_{\mathcal{B}}(\text{PrivK}^{\text{eav}}_{\mathcal{B},\Pi}(n, 1)) = 1\right] &= \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1 | b = 1\right]
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
&\Pr\left[\text{out}_{\mathcal{B}}(\text{PrivK}^{\text{eav}}_{\mathcal{B},\Pi}(n, 0)) = 1\right] - \Pr\left[\text{out}_{\mathcal{B}}(\text{PrivK}^{\text{eav}}_{\mathcal{B},\Pi}(n, 1)) = 1\right] \\
&= \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 0 | b = 0\right] - \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1 | b = 1\right] \\
&= 1 - \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1 | b = 0\right] - pr\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1 | b = 1 \\
&= 1 - 2(\frac{1}{2} + \epsilon) \\
&= -2\epsilon
\end{aligned}$$

Thus π is not IND-EAV2.

Example 3.3 OTP is IND-EAV.

3.4 Computationally Indistinguishable

Definition 3.10 Let R be a finite or countable set. Let X, Y be two random variables that take values in R. The statistical distance between X and Y is defined as

$$\mathrm{SD}(X,Y) = \frac{1}{2}\sum_{r\in R} |\Pr[X=r] - \Pr[Y=r]|.$$

Example 3.4 Define the following random variables X, Y:

- X : $\Pr[X=0^n]=0$ and $\Pr[X=r]=\frac{1}{2^n-1}$ for all $r\in\{0,1\}^n\backslash 0^n$.
- Y : $\Pr[Y=r]=\frac{1}{2^n}$ for every $r\in\{0,1\}^n$.

The statistical distance between X and Y is

$$\mathrm{SD}(X,Y) = \frac{1}{2}\left(\left|0-\frac{1}{2^n}\right| + (2^n-1)\left|\frac{1}{2^n-1}-\frac{1}{2^n}\right|\right) = \frac{1}{2^n}$$

Definition 3.11 Let X_n, Y_n be random varibales for $n=1,2,\cdots$. We call $X=\{X_n\}$ and $Y=\{Y_n\}$ probability ensembles. Then X and Y are said to be computationally indistinguishable if for any PPT distinguisher $\mathcal{D}$, there is a negligible function $\mathrm{negl}(n)$ such that

$$|\Pr[\mathcal{D}(1^n, X_n)=1] - \Pr[\mathcal{D}(1^n, Y_n)=1]| \le \mathrm{negl}(n)$$

Example 3.5 Let $X=\{X_n\}_{n\ge 1}, Y=\{Y_n\}_{n\ge 1}$, each ensemble in X and Y defined as follows.

- X_n : $\Pr[X_n=0^n]=0$ and $\Pr[X_n=r]=\frac{1}{2^n-1}$ for all $r\in\{0,1\}^n\backslash 0^n$.
- Y_n : uniformly random variable over $\{0,1\}^n$.

How to determine they are whether computationally indistinguishable?

Note We use p.i. for perfectly indistinguishable, s.i. for statistically indistinguishable and c.i. for computationally indistinguishable.

Theorem 3.7 $X \equiv_{\text{p.i}} Y \Rightarrow X \equiv_{\text{s.i}} Y \Rightarrow X \equiv_{\text{c.i}} Y$.

Theorem 3.8 $X \equiv_a Y$ *for* $a \in \{\text{p.i}, \text{s.i}, \text{c.i}\}$.

Theorem 3.9 $X \equiv_a Y \Rightarrow Y \equiv_a X$ *for* $a \in \{\text{p.i}, \text{s.i}, \text{c.i}\}$.

Theorem 3.10 $X \equiv_a Y$ *and* $Y \equiv_b Z$, *then* $X \equiv_{\text{weak}(a,b)} Z$, *where*

$$\text{weak}(a,b) = \begin{cases} a & \text{if } b \Rightarrow a \\ b & \text{if } a \Rightarrow b \end{cases}$$

for all $a, b \in \{\text{p.i}, \text{s.i}, \text{c.i}\}$

3.5 Pseudorandomness

Definition 3.12 Let $X=\{X_n\}$ be probability ensemble, where the X_n is distributed over $\{0,1\}^n$. Let U_n be uniformly distributed over $\{0,1\}^n$. The probability ensemble X is said to be pseudorandom if X and $U=\{U_n\}$ are computationally indistinguishable.

Example 3.6 X in Example 3.5 is a pseudorandom.

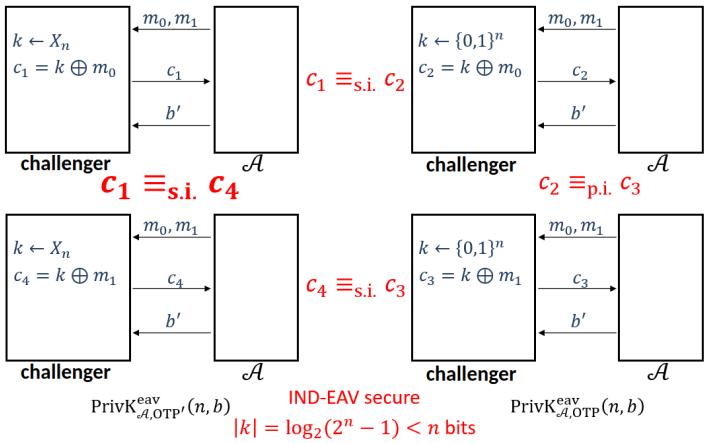


Figure 6: IND-EAV from Pseudorandomness

Theorem 3.11 *We can use the X_n instead of U_n in OTP, the new scheme is still IND-EAV.*

The key can reduce to $n-1$ bits $\log_2(2^n - 1)$.

Definition 3.13 A function $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ is called pseudorandom generator (PRG) if it satisfies the following conditions:

- efficient computable: there is a deterministic polynomial-time algorithm D such that $D(x) = G(x)$ for all $x \in \{0,1\}^n$.
- expansion: $l(n) > n$ for all $n \in N$, where l is called the expandsion factor of G.
- pseudorandomness: $G(U_n)$ is pseudorandom, where U_n is uniformly distributed over $\{0,1\}^n$, that is for any PPT algorithm distinguisher $\mathcal{D}$, exists a negligible function negl$(\cdot)$ such that

$$|\Pr[\mathcal{D}(G(U_n)) = 1] - \Pr[\mathcal{D}(U_{l(n)}) = 1]| \leq \text{negl}(n).$$

Example 3.7 Let function $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ defined by

$$G(s_1 s_2 \cdots s_n) = s_1 s_2 \cdots s_n (s_{n+1}),$$

where $s_{n+1} = s_1 \oplus s_2 \oplus \cdots \oplus s_n$. We could construct a adversary show G is not PRG as follows.

- get x from $G(U_n)$ or U_{n+1}.
- output 1 if $x_{n+1} = x_1 \oplus x_2 \oplus \cdots \oplus x_n$; otherwise, output 0.

We show two probabilities

$$\Pr[\mathcal{D}(G(U_n)) = 1] = \Pr_{x \leftarrow G(U_n)}[D(x) = 1] = 1$$

$$\Pr[\mathcal{D}(U_{n+1}) = 1] = \Pr_{x \leftarrow U_{n+1}}[D(x) = 1] = \Pr_{x \leftarrow U_{n+1}}[x_{n+1} = \oplus_{i=1}^{n} x_i] = \frac{1}{2} \quad (x_{n+1} = 0 \text{ or } x_{n+1} = 1)$$

Thus $|\Pr[\mathcal{D}(G(U_n)) = 1] - \Pr[\mathcal{D}(U_{n+1}) = 1]| = \frac{1}{2}$ is non-negligible.

Example 3.8 Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a PRG. We show $G(U_n)$ is far from uniform. Let Im(G) be the

image $G(1^n)$. Then we have $\text{Im}(G) \leq 2^n$, far less than 2^{2n}. Consider the $\text{SD}(G(U_n), U_{2n})$:

$$\begin{aligned}
\text{SD}(G(U_n), U_{2n}) &= \frac{1}{2} \sum_{r=\{0,1\}^{2n}} |\Pr[G(U_n) = r] - \Pr[U_{2n} = r]| \\
&\geq \frac{1}{2} \sum_{r \notin \text{Im}(G)} |\Pr[G(U_n) = r] - \Pr[U_{2n} = r]| \\
&\geq \frac{1}{2} \cdot |0 - \frac{1}{2^{2n}}| \cdot (2^{2n} - 2^n) \\
&= \frac{1}{2} - \frac{1}{2^{n+1}}
\end{aligned}$$

Remark The seed s for G must be long enough for preventing bruth-force attack. In addition, the seed s must be chosen uniformly and kept secret.

Example 3.9 (Fixed-length Encryption from PRG) Let $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ be a PRG. Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ + $\mathcal{M}$ is defined as follows.

- $\text{Gen}(1^n)$: choose $k \leftarrow \{0,1\}^n$ and output k.
- $\text{Enc}(k, m)$: output $c = G(k) \oplus m$.
- $\text{Dec}(k, c)$: output $G(k) \oplus c$.

Theorem 3.12 *The scheme Π is IND-EAV secure.*

Proof. Suppose Π is not IND-EAV secure. Then exists a PPT adversary $\mathcal{A}$ such that

$$\epsilon = \left| \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2} \right|$$

is non-negligible in n. Without loss of generality, assume that

$$\Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \geq \frac{1}{2}.$$

We give a contradiction that show G is not a PRG. We could construct a distinguisher $\mathcal{D}$ combined with $\mathcal{A}$ in Figure 7. Distinguisher $\mathcal{D}$ take the result of the correctness of $\mathcal{A}$ outputs as output. Consider two cases:

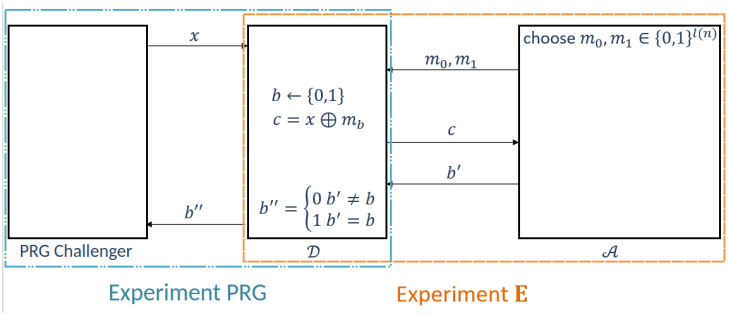


Figure 7: OTP from Pseudorandomness

- When $x \leftarrow G(U_n)$, then the experiment E is $\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n)$. We have

$$\Pr[\mathcal{D} = 1] = \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] = \frac{1}{2} + \epsilon.$$

- When $x \leftarrow U_{n+1}$, then the experiment E is $\text{PrivK}^{\text{eav}}_{\mathcal{A},\text{OTP}}(l(n))$. We have

$$\Pr[\mathcal{D} = 1] = \Pr\left[\text{PrivK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] = \frac{1}{2}.$$

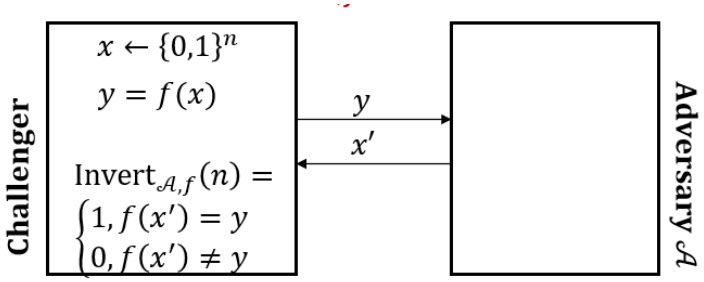


Figure 8: Inverting experiment

Thus $|\Pr[\mathcal{D}(G(U_n)) = 1] - \Pr[\mathcal{D}(U_{n+1}) = 1]| = \epsilon$ is non-negligible.

Definition 3.14 The inverting experiment $\text{Invert}_{\mathcal{A},f}(n)$ for $f : \{0,1\}^* \to \{0,1\}^*$ is defined in Figure 8.

Definition 3.15 $f : \{0,1\}^* \to \{0,1\}^*$ is a one-way function (OWF) if

- eazy to compute: exist DPT C such that $C(x) = f(x)$ for all $x \in \{0,1\}^*$.
- hard to invert: for all PPT $\mathcal{A}$, exist a negligible function $\text{negl}(\cdot)$ such that

$$\Pr[\text{Invert}_{\mathcal{A},f}(n) = 1] \leq \text{negl}(n)$$

One-way permutation (OWP): $f : \{0,1\}^n \to \{0,1\}^n$ is bijiective for all n.

Example 3.10 (OWP) $f_{p,g}(x) = g^x \mod p$ is conjectured as a OWP, where p is a prime, g is a generator of $\mathbb{Z}_p^*$ and $x \in \{1, 2, \cdots, p-1\}$.

Definition 3.16 Let $f : \{0,1\}^* \to \{0,1\}^*$ and $\text{hc} : \{0,1\}^* \to \{0,1\}$ be two functions. hc is a hard-core predicate (HCP) for f if

- easy to compute: exist DPT C such that $C(x) = \text{hc}(x)$ for every $x \in \{0,1\}^*$.
- hard to predicate: for all PPT $\mathcal{A}$, exist a negligible function $\text{negl}(\cdot)$ such that

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(1^n, f(x)) = \text{hc}(x)] \leq \frac{1}{2} + \text{negl}(n)$$

Theorem 3.13 *(Goldreich-Levin theorem) Assume that OWFs (OWPs) exist. Then there is a OWF (OWP) g and a hand-core predicate* hc *for g.*

Theorem 3.14 *(PRG from HCP) Let f be a OWP and let* $\text{hc}(x)$ *be a hard-core predicate for f. Then*

$$G(x) = (f(x), hc(x)) : \{0,1\}^n \to \{0,1\}^{n+1}$$

is a PRG.

Example 3.11 (PRG with arbitrary expansion) Let f be a OWP and let $\text{hc}(x)$ be a hard-core predicate. Construct a PRG $G' : \{0,1\}^n \to \{0,1\}^{n+p(n)}$ in Figure 9.

3.6 IND-m-EAV

Definition 3.17 Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ be a private-key encryption. Define an multiple-message eavesdropping experiment $\text{PrivK}^{\text{mult}}_{\mathcal{A},\Pi}$ in Figure 10.

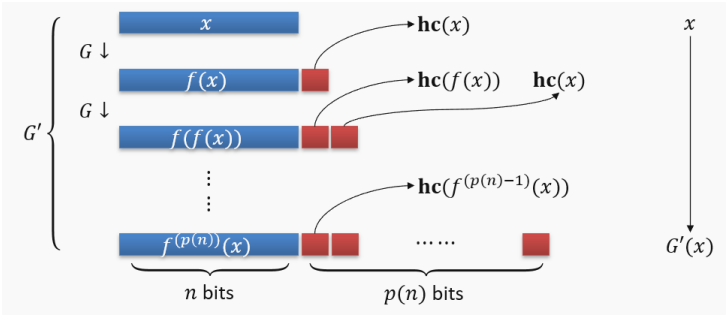


Figure 9: PRG with arbitrary expansion

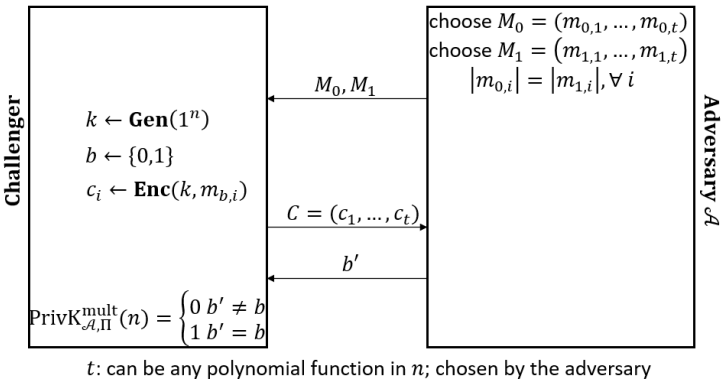


Figure 10: Multiple-message eavesdropping experiment

Definition 3.18 Π has indistinguishable multiple encryptions in the presence of an eavesdropper (IND-m-EAV) if for all PPT adversary $\mathcal{A}$ there is a negligible function $\text{negl}(\cdot)$ such that

$$\Pr\left[\text{PrivK}^{\text{mult}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \text{negl}(n).$$

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment. There is equivalent definition

$$\left|\Pr\left[\text{PrivK}^{\text{mult}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2}\right| \leq \text{negl}(n).$$

Note IND-EAV is special case of IND-m-EAV.

Example 3.12 OTP is IND-EAV secure but not IND-m-EAV secure. We can construct a adversary as follows

- let $M_0 = (0^n, 0^n), M_1 = (0^n, 1^n)$.
- learn $C = (c_1, c_2)$, output 0 if $c_1 = c_2$; otherwise output 1.

Theorem 3.15 *If* Π *is stateless and* Enc *is deterministic, then* Π *cannot be IND-m-EAV secure.*

Stateless means that each invocation of Enc is independent of all prior invocations. Deterministic means that Enc will always return same result for same plain-text and key. Thus Π must be stateful or probabilistic if Π is IND-m-EAV.

3.7 IND-CPA

Definition 3.19 Chosen plaintext attack indistinguishability experiment $\text{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n)$ is defined in Figure 12.

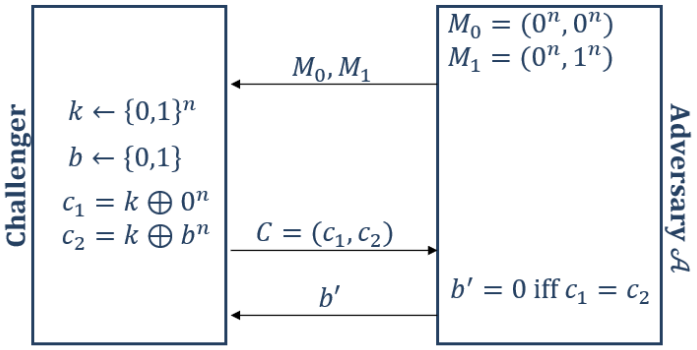


Figure 11: OTP is not IND-m-EAV

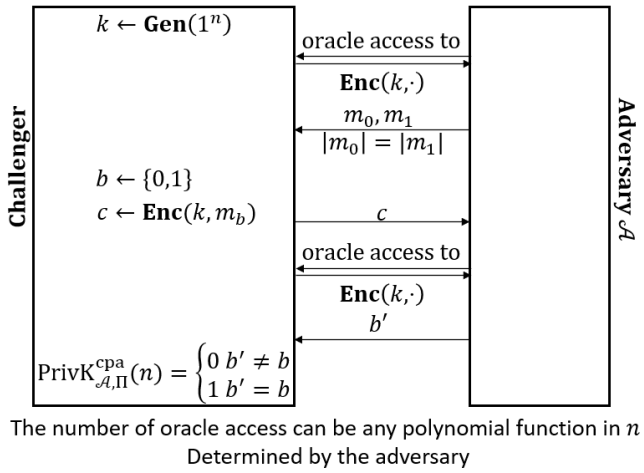


Figure 12: CPA indistinguishability experiment

Definition 3.20 Π has indistinguishable encryptions under a chosen-plaintext attack (IND-CPA) if for all PPT adversary $\mathcal{A}$ there is a negligible function $\text{negl}(\cdot)$ such that

$$\Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \text{negl}(n),$$

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment. There is an equivalent definition

$$\left|\Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2}\right| \leq \text{negl}(n)$$

Definition 3.21 Let $\text{LR}_{k,b}(\cdot)$ be that given $m_0, m_1 \in \mathcal{M}$ and output $c \leftarrow \text{Enc}(k, m_b)$. LR-Orcale experiment $\text{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n)$ is defined in Figure 13.

Note Instead of outputting the list $(m_{0,1}, \cdots, m_{0,t})$ and $(m_{1,1}, \cdots, m_{1,t})$, one of whose messages will be encrypted, the attacker can query $\text{LR}_{k,b}(m_{0,1}, m_{1,1}), \cdots, \text{LR}_{k,b}(m_{0,t}, m_{1,t})$. This also encompasses the attacker's access to an encryption orcale, since the attacker can simply query $\text{LR}_{k,b}(m, m)$ to obtain $\text{Enc}_k(m)$.

Definition 3.22 Π has indistinguishable multiple encryptions under a chosen-plaintext attack (IND-m-CPA) if for all PPT adversary $\mathcal{A}$ there is a negligible function $\text{negl}(\cdot)$ such that

$$\Pr\left[\text{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \text{negl}(n).$$

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment. There is an equivalent definition. There is an equivalent definition:

$$\left|\Pr\left[\text{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2}\right| \leq \text{negl}(n)$$

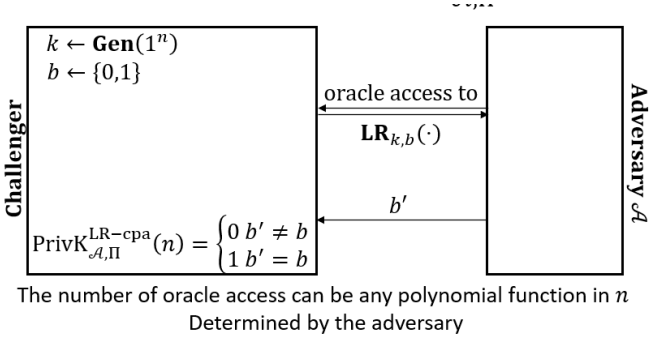


Figure 13: LR-CPA indistinguishability experiment

Remark There are three implications:

- IND-m-CPA secure $\Rightarrow$ IND-CPA
- IND-m-CPA secure $\Rightarrow$ IND-m-EAV
- IND-CPA secure $\Rightarrow$ IND-m-CPA secure (not trivial)

Example 3.13 (Fixed-length scheme to arbitrary-length) Let $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec}) + \mathcal{M}$ is IND-CPA secure. we can construct a new scheme $\Pi' = (\mathrm{Gen}', \mathrm{Enc}', \mathrm{Dec}') + \mathcal{M}'$ as follows.

- $\mathrm{Gen}'(1^n)$: output $k \leftarrow \mathrm{Gen}(1^n)$.
- $\mathrm{Enc}'(k, m)$: let $m = m_1 m_2 \cdots m_t \in \mathcal{M}^t$, for each $c_i = \mathrm{Enc}(k, m_i)$ $(i \in [t])$, output $c = c_1 c_2 \cdots c_t$.
- $\mathrm{Dec}'(k, c)$: let $c = c_1 c_2 \cdots c_t$, for each $m_i = \mathrm{Dec}(k, c_i)$ $(i \in [t])$, output $m = m_1 m_2 \cdots m_t$.

Theorem 3.16 *Π' in Example 3.13 is IND-CPA.*

Corollary *If Π in Example 3.13 is IND-m-CPA, then Π' is IND-m-CPA.*

3.8 Pseudorandom Function

Definition 3.23 A keyed function is a two-input function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, where the first input k is called key. We simply use the notation $F_k(x)$ for $F(k, x)$. We say F is efficient if there exists DPT algorithm computing $F(k, x)$. F is length-preserving if $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$.

Definition 3.24 Let F be a length preserving keyed function. For every n, F induces a distribution F_k over $\mathrm{Func}_n = \{f | f : \{0,1\}^n \to \{0,1\}^n\}$, there set of all functions that map n-bit input to n-bit output: choose $k \to \{0,1\}^n$ uniformly and at random, output F_k.

Definition 3.25 The PRF distingushability experiment is defined in Figure 14, where $k \to \{0,1\}^*$ and $\mathcal{O}$ random

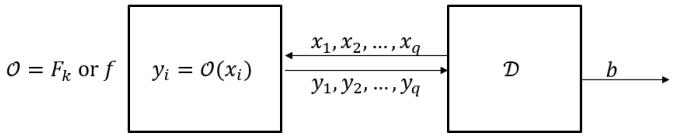


Figure 14: PRF indistinguishability experiment

taken from F_k and f.

Definition 3.26 Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed function. F is said

to be a PRF if for all PPT algorithm $\mathcal{D}$, there is a negligible function $\text{negl}(\cdot)$ such that

$$\left|\Pr\left[\mathcal{D}^{F_k(\cdot)}=1\right]-\Pr\left[\mathcal{D}^{f(\cdot)}=1\right]\right|\leq \text{negl}(n)$$

Example 3.14 Let the keyed function $F:\{0,1\}^n\times\{0,1\}^n\to\{0,1\}^n$ be defined such that $F(k,x)=k\oplus x$ for all $k\in\{0,1\}^n$ and $x\in\{0,1\}^n$. We could construct an adversary show F is not PRF.

- choose two distinct $x,x'\leftarrow\{0,1\}^n$ as inputs for oracle.
- learns y,y' from oracle, if $y\oplus y'=x\oplus x'$, then output 1; otherwise, output 0.

Consider two cases:

- when $\mathcal{O}=F_k$, we have $\Pr\left[\mathcal{D}^{F_k(\cdot)}=1\right]=1$.
- when $\mathcal{O}=f$, we have $\Pr\left[\mathcal{D}^{f(\cdot)}=1=1\right]=\frac{1}{2^n}$.

Thus $\left|\Pr\left[\mathcal{D}^{F_k(\cdot)}=1\right]-\Pr\left[\mathcal{D}^{f(\cdot)}=1\right]\right|=1-\frac{1}{2^n}$ is non-negligible.

Example 3.15 (IND-CPA from PRF) Let F is a length-preserving PRF. Let $\Pi=(\text{Gen},\text{Enc},\text{Dec})+\mathcal{M}$ be a fixed-length scheme defined as follows.

- $\text{Gen}(1^n)$: choose $k\leftarrow\{0,1\}^n$ and output k.
- $\text{Enc}(k,m)$: let $m\in\{0,1\}^n$, choose $r\leftarrow\{0,1\}^n$ and output $c=(r,F_k(r)\oplus m)$.
- $\text{Dec}(k,c)$: let $c=(r,s)$, output $s\oplus F_k(r)$.

Theorem 3.17 *The scheme Π in Example 3.15 is IND-CPA secure.*

Proof. Firstly, we construct a scheme $\Pi'=(\text{Gen}',\text{Enc}',\text{Dec}')+\mathcal{M}$ that use $f\leftarrow\text{Func}_n$ to substitute the use of F_k as follows.

- $\text{Gen}(1^n)$: choose $f\leftarrow\text{Func}_n$ and output $k=f$.
- $\text{Enc}(k,m)$: let $m\in\{0,1\}^n$, choose $r\leftarrow\{0,1\}^n$ and output $c=(r,f(r)\oplus m)$.
- $\text{Dec}(k,c)$: let $c=(r,s)$, output $s\oplus f(r)$.

Secondly, we show that

$$\epsilon_1=\left|\Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n)=1\right]-\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n)=1\right]\right|$$

is negligible. Thirdly, we show that

$$\epsilon_2=\left|\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n)=1\right]-\frac{1}{2}\right|$$

is negligible. Finally, by the triangle inequality we have $\left|\Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n)=1\right]-\frac{1}{2}\right|\leq\epsilon_1+\epsilon_2$.

For the first inequality, we suppose ϵ_1 is non-negligible. we show there is a PPT algorithm $\mathcal{D}$ that distinguishes between F_k and $f\leftarrow\text{Func}_n$ that is a contradiction. We can construct a distinguisher in Figure 15. Consider two cases:

$$\begin{aligned}\Pr\left[\mathcal{D}^{F_k(\cdot)}=1\right]&=\Pr\left[b''=1|\mathcal{O}=F_k\right]\\&=\Pr\left[b'=b|\mathcal{O}=F_k\right]\\&=\Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n)=1\right]\\\Pr\left[\mathcal{D}^{f(\cdot)}=1\right]&=\Pr\left[b''=1|\mathcal{O}=f\right]\\&=\Pr\left[b'=b|\mathcal{O}=f\right]\\&=\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n)=1\right]\end{aligned}$$

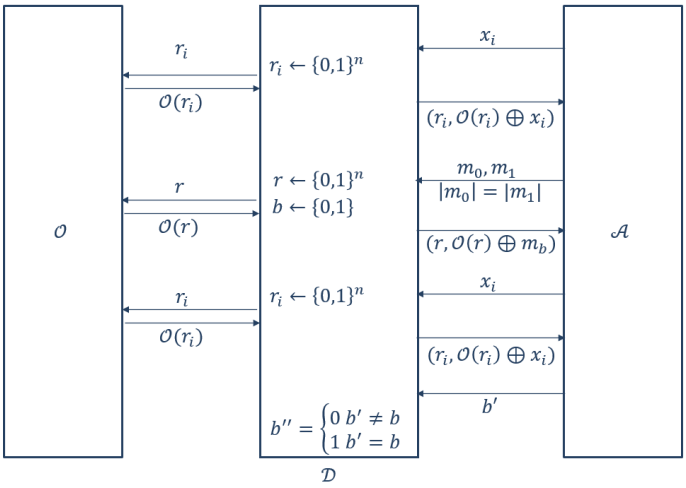


Figure 15: Counterexample distinguisher for Π and Π'

Thus we have

$$\left|\Pr\left[\mathcal{D}^{F_k(\cdot)} = 1\right] - \Pr\left[\mathcal{D}^{f(\cdot)} = 1\right]\right| = \left|\Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right]\right| = \epsilon$$

is non-negligible.

For the second inequality, we show ϵ_2 is negligible straightly. Consider the IND-CPA security experiment Π' in Figure Suppose we have queried $q = q(n)$ times, each query corresponding a r_i ($i \in [1, q]$). For each query, we

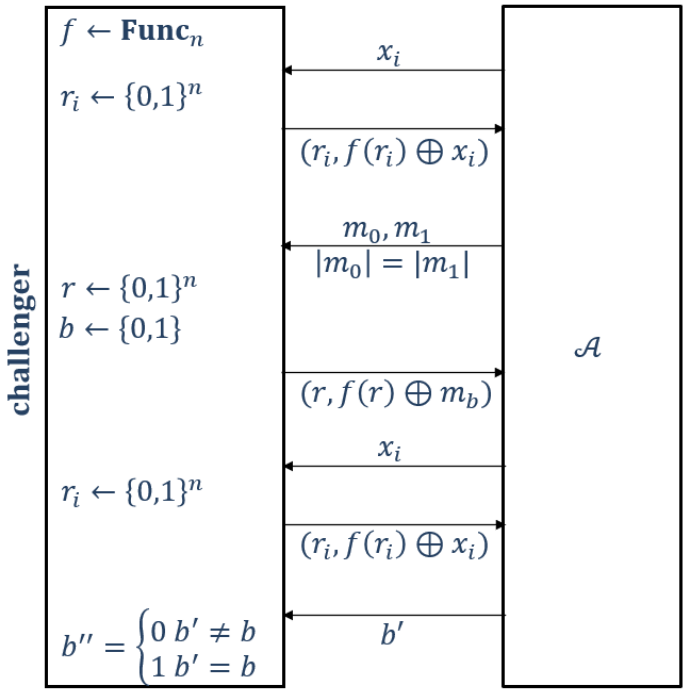


Figure 16: CPA experiment for Π'

could get $f(r_i)$ by $(f(r_i) \oplus x_i) \oplus x_i$. Then We consider two cases:

- if $r \in \{r_1, \cdots, r_q\}$, then $f(r) = f(r_j)$ for some $j \in [1, q]$, we can get m_b immediately by $m_b = (f(r) \oplus m_b) \oplus f(r)$. Thus $\Pr[b' = b] = 1$ in this case.
- if $r \notin \{r_1, \cdots, r_q\}$, then $f(r) \oplus m_b$ is truely random, so $\Pr[b' = b] = \frac{1}{2}$ in this case.

We use RP for the event that $r \in \{r_1, \cdots, r_q\}$, it is clear that $\Pr[\text{RP}] \leq \frac{q}{2^n}$. Since some r_i are possibly same one. Finally, we have

$$\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right] = \Pr[\text{RP}] + \Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1 \wedge \neg\text{RP}\right] \leq \frac{q}{2^n} + \frac{1}{2}$$

From perspective of above cases, it trivailly show that $\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right] \geq \frac{1}{2} \geq \frac{1}{2} - \frac{q}{2^q}$. Since the probability under each case is large than $\frac{1}{2}$. Thus we have ϵ_2 is negligible.

Example 3.16 (PRF from PRG) Given a PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$, we could construct a length-preserving PRF $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ as follows.

- Let $G(k) = G_0(k)||G_1(k)$, where $|G_0(k)| = |G_1(k)|$.
- Let $F(k,x) = G_{x_n}(G_{x_{n-1}}(\cdots(G_{x_2}(G_{x_1}(k)))\cdots))$, where $x = x_1x_2\cdots x_n$ that each x_i is bit.

For $n = 3$, it show in Figure 17.

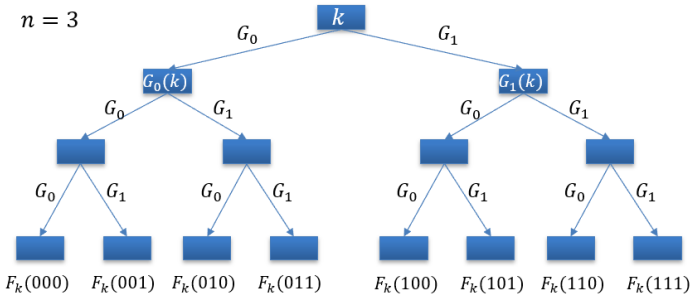


Figure 17: PRF from PRG

Example 3.17 (PRG from PRF) Given a length preserving PRF $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, we could construct a PRG $G : \{0,1\}^n \to \{0,1\}^{ln}$ such that

$$G(k) = F_k(1)||F_k(2)||\cdots||F_k(l)$$

3.9 Pseudorandom Permutation

Definition 3.27 A keyed permutation $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a two-input function such that $F_k : \{0,1\}^n \to \{0,1\}^n$ is bijective for every $k \in \{0,1\}^n$.

Definition 3.28 Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient computable, efficiently invertible, length-perserving, keyed permutation. F is said to be a pseudorandom permuatation (PRP) if for all PPT algorithm $\mathcal{D}$, there is a negligible function $\text{negl}(\cdot)$ such that

$$\left|\Pr\left[\mathcal{D}^{F_k(\cdot)} = 1\right] - \Pr\left[\mathcal{D}^{f(\cdot)} = 1\right]\right| \leq \text{negl}(n)$$

where the probabilities are taken over $k \leftarrow \{0,1\}^n$, $f \leftarrow \text{Perm}_n$ and randomness of $\mathcal{D}$.

Theorem 3.18 *If F is a PRP and $l_{\text{in}}(n) \geq n$, then F is a PRF.*

Definition 3.29 The distinguishability experiment for sPRP is defined in Figure 18. Note there are oracle for two side of F_k or f.

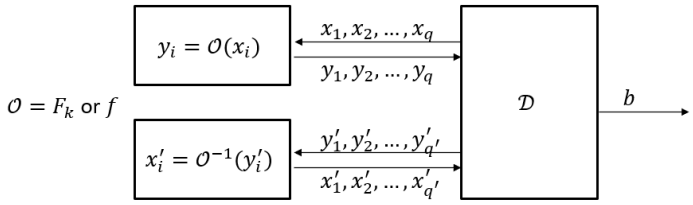


Figure 18: Distinguishability experiment for sPRP

Definition 3.30 Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient computable, efficiently invertible, length-perserving, keyed permutation. F is said to be a strong pseudorandom permuatation (sPRP) if for all PPT algorithm $\mathcal{D}$, there is a negligible function negl$(\cdot)$ such that

$$\left|\Pr\left[\mathcal{D}^{F_k(\cdot),F_k^{-1}(\cdot)} = 1\right] - \Pr\left[\mathcal{D}^{f(\cdot),f^{-1}(\cdot)} = 1\right]\right| \leq \text{negl}(n)$$

where the probabilities are taken over $k \leftarrow \{0,1\}^n$, $f \leftarrow \text{Perm}_n$ and randomness of $\mathcal{D}$.

Example 3.18 (PRP from PRF) Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a length-preserving PRF. We could construct a $P : \{0,1\}^{3n} \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ as follows.

- Key k: let $k = (k_1, k_2, k_3) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$.
- Input x: let $x = L_0 || R_0 \in \{0,1\}^n \times \{0,1\}^n$.
- Output y let $y = L_3 || R_3 \in \{0,1\}^n \times \{0,1\}^n$, where L_3 and R_3 are calculated by a 3-round Feistel network:
  - $L_1 = R_0$, $R_1 = L_0 \oplus F_{k_1}(R_0)$.
  - $L_2 = R_1$, $R_2 = L_1 \oplus F_{k_1}(R_1)$.
  - $L_3 = R_2$, $R_3 = L_2 \oplus F_{k_1}(R_2)$.

Theorem 3.19 *The P in Example 3.18 is a PRP.*

3.10 Models of Operation

The cipher is often large than plaintext. For example, the PRF-based encryption scheme implies $|c| = 2|m|$ that the cipher is twice as long as message. Choosing the proper mode of operation, the problem could solved safety. There are four modes:

- **ECB**: Electronic Code Book mode.
- **CBC**: Cipher Block Chaining mode.
- **OFB**: Output Feedback mode.
- **CTR**: Counter mode.

We have to partition the plaintext into blocks when using these modes. However the length of plaintext may not to be a multiple of block length, we have to use padding techniques. In addition, we aslo have to add a block in the end when the length of plaintext is a multiple of block length for decryption consistency.

Definition 3.31 (ECB) Let F is block cipher with block length n. Define a encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ as follows.

- Gen(1^n): let $k \leftarrow \{0,1\}^n$ and output k.
- Enc(k, m): let $m \in \{0,1\}^{tn}$ and $m = m_1 \cdots m_t$.
  - Let $c_i = F_k(m_i)$ for all $i \in [t]$.

- Output $c = c_1 \cdots c_t$.

- Dec(k, c): let $c \in \{0,1\}^{(t)n}$ and $c = c_0 c_1 \cdots c_t$.
  - Let $m_i = F_k^{-1}(c_i)$ for all $i \in [t]$.
  - Output $m = m_1 m_2 \cdots m_t$.

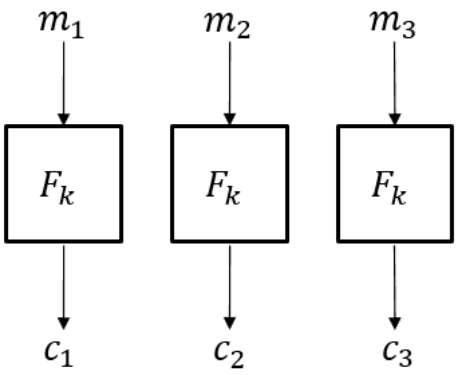


Figure 19: ECB mode

Note ECB is deterministic, it is even not IND-EAV secure!

Definition 3.32 (CBC) Let F is block cipher with block length n. Define a encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})+\mathcal{M}$ as follows.

- Gen(1^n): let $k \leftarrow \{0,1\}^n$ and output k.
- Enc(k, m): let $m \in \{0,1\}^{tn}$ and $m = m_1 \cdots m_t$.
  - Let $IV \leftarrow \{0,1\}^n$ and $c_0 \leftarrow IV$.
  - Let $c_i = F_k(c_{i-1} \oplus m_i)$ for all $i \in [t]$.
  - Output $c = c_0 c_1 \cdots c_t$.
- Dec(k, c): let $c \in \{0,1\}^{(t+1)n}$ and $c = c_0 c_1 \cdots c_t$.
  - Let $m_i = F_k^{-1}(c_i) \oplus c_{i-1}$ for all $i \in [t]$.
  - Output $m = m_1 m_2 \cdots m_t$.

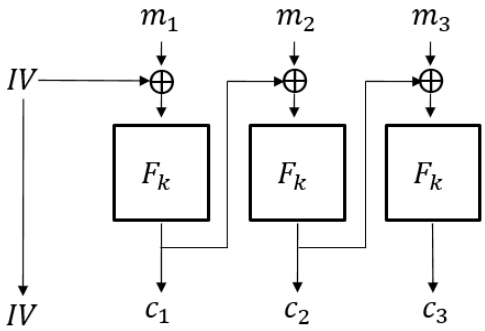


Figure 20: CBC mode

Theorem 3.20 *If F is PRP, then CBC is IND-CPA secure.*

Note The IV in CBC must be chosen uniformly at random.

Example 3.19 If we use $IV = i$ to encrypt the ith plaintext, then the new scheme is not IND-CPA secure. We could construct an adversary as follows.

- send $x = 0^{n-1}1$ to oracle, and get $y = (1, F_k(1 \oplus x)) = (1, F_k(0^n))$.
- send $m_0 = 0^{n-2}10, m_1 = 0^n$ to challenger, and learns $c = (2, F_k(2 \oplus m_b)) = (\alpha, \beta)$.
- if $\beta = F_k(0^n)$, then output 0; otherwise output 1.

Example 3.20 The chained CBC was purposed for reduce the bandwidth used in SSL3.0 and TLS 1.0, but it not IND-CPA secure. It looks like in Figure 21. We could construc an adversary as follows.

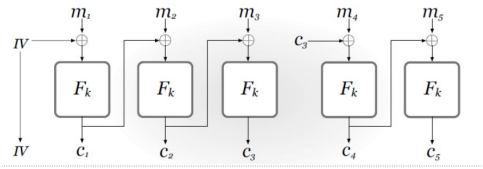


Figure 21: Chained ECB mode

- send $m_0 = a_0x_2x_3$ and $m_1 = a_1x_2x_3$ to challenger, and learns $y = c_0c_1c_2c_3$, where $c_1 = F_k(c_0 \oplus a_b)$.
- send $m_4 = c_0 \oplus a_0 \oplus x_3$, then $c_4 = F_k(c_0 \oplus a_0)$.
- if $c_4 = c_1$, then output 0; otherwise output 1.

Definition 3.33 (OFB) Let F is block cipher with block length n. Define a encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ as follows.

- $\text{Gen}(1^n)$: let $k \leftarrow \{0,1\}^n$ and output k.
- $\text{Enc}(k, m)$: let $m \in \{0,1\}^{tn}$ and $m = m_1 \cdots m_t$.
  - Let $IV \leftarrow \{0,1\}^n$ and $c_0, y_0 \leftarrow IV$.
  - Let $y_i = F_k(y_{i-1})$ and $c_i = m_i \oplus y_i$ for all $i \in [t]$.
  - Output $c = c_0c_1 \cdots c_t$.
- $\text{Dec}(k, c)$: let $c \in \{0,1\}^{(t+1)n}$ and $c = c_0c_1 \cdots c_t$.
  - Let $y_i = F_k(y_{i-1})$ and $m_i = y_i \oplus c_i$ for all $i \in [t]$.
  - Output $m = m_1m_2 \cdots m_t$.

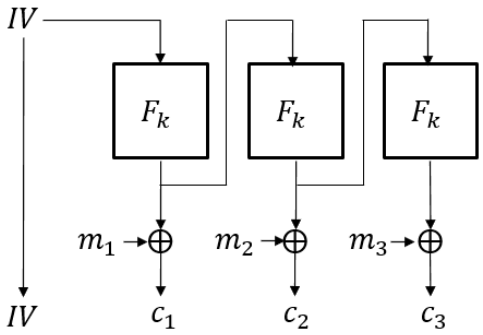


Figure 22: OFB mode

Definition 3.34 (CTR) Let F is a block cipher with block length n. Define a encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ as follows.

- $\text{Gen}(1^n)$: let $k \leftarrow \{0,1\}^n$ and output k.
- $\text{Enc}(k, m)$: let $m \in \{0,1\}^{tn}$ and $m = m_1 \cdots m_t$.
  - Let ctr $\rightarrow \{0,1\}$, $c_0 = \text{ctr}$ and $c_i = F_k(\text{ctr} + i) \oplus m_i$ for $i \in [t]$.
  - Output $c = c_0c_1 \cdots c_t$.

- $\mathrm{Dec}(k, c)$: let $c \in \{0,1\}^{(t+1)n}$ and $c = c_0 c_1 \cdots c_t$.
  - Let $m_i = F_k(c_0 + i) \oplus c_i$ for all $i \in [t]$.
  - Output $m = m_1 m_2 \cdots m_t$.

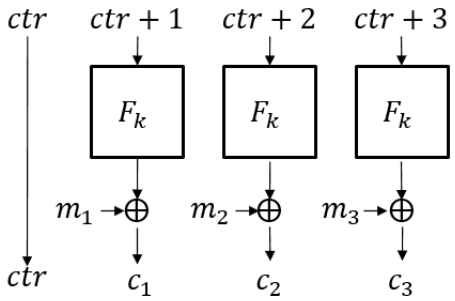


Figure 23: CTR mode

Note ctr can never be reused; ctr will repeat after $2^{n/2}$ encrytions, thus n cannot be too small. The advantage of CTR mode is that the calculation can be in parallel.

Lemma 3.21 *Let* $\boldsymbol{Func}_n$ *is the set of all possible* n *length-preserving function. Defines a encryption scheme* $\Pi' = (\mathrm{Gen}', \mathrm{Enc}', \mathrm{Dec}')$ *as follows.*

- $\mathrm{Gen}'(1^n)$*: let* $f \to \boldsymbol{Func}_n$ *and output* f.
- $\mathrm{Enc}'(k, m)$*: let* $m \in \{0,1\}^{tn}$ *and* $m = m_1 \cdots m_t$.
  - *Let* $\mathrm{ctr} \to \{0,1\}$, $c_0 = \mathrm{ctr}$ *and* $c_i = f(\mathrm{ctr} + i) \oplus m_i$ *for* $i \in [t]$.
  - *Output* $c = c_0 c_1 \cdots c_t$.
- $\mathrm{Dec}'(k, c)$*: let* $c \in \{0,1\}^{(t+1)n}$ *and* $c = c_0 c_1 \cdots c_t$.
  - *Let* $m_i = f(c_0 + i) \oplus c_i$ *for all* $i \in [t]$.
  - *Output* $m = m_1 m_2 \cdots m_t$.

Then we have

$$\left|\Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right]\right| \leq \mathrm{negl}(n)$$

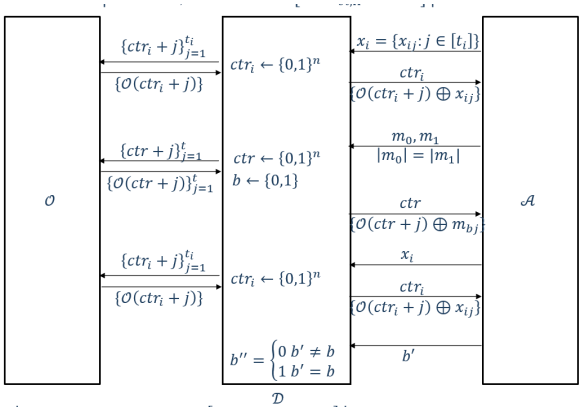


Figure 24: Reduction of CTR

Proof. Assume $\left|\Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right]\right| > \mathrm{negl}(n)$, wlg. we assume

$$\Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right] > \mathrm{negl}(n)$$

This is we can construct a adversary $\mathcal{A}$ that the probability of guess right in Π is large than Π'. The $\mathcal{A}$ in Figure 24 is a such adversary for both Π and Π'. The distinguisher $\mathcal{D}$ is a part of for PRF F testing experiment. The oracle $\mathcal{O}$ is either F_k or $f \in \mathbf{Func}_n$. Notice there lacks one last edge that $\mathcal{D}$ output b'' to oracle $\mathcal{O}$.

We have to make things clear. For $\mathcal{O}$, there are two cases:

- If $\mathcal{O} = F_k$, then $\mathcal{A}$ is a part of Π experiment.
- If $\mathcal{O} = f$, then $\mathcal{A}$ is a part of Π' experiment.

From above two cases, we can calculate two critical probability of $\mathcal{D}$. For the first, we have

$$\begin{aligned}\Pr\left[\mathcal{D}^{F_k} = 1\right] &= \Pr\left[b'' = 1 | \mathcal{O} = F_k\right] \\ &= \Pr\left[b' = b | \mathcal{O} = F_k\right] \qquad \text{(We are in } \Pi \text{ experiment now)} \\ &= \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n) = 1\right]\end{aligned}$$

For the second, we have

$$\begin{aligned}\Pr\left[\mathcal{D}^{f} = 1\right] &= \Pr\left[b'' = 1 | \mathcal{O} = f\right] \\ &= \Pr\left[b' = b | \mathcal{O} = f\right] \qquad \text{(We are in } \Pi' \text{ experiment now)} \\ &= \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right]\end{aligned}$$

Thus we have

$$\Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right] = \Pr\left[\mathcal{D}^{F_k} = 1\right] - \Pr\left[\mathcal{D}^{f} = 1\right] > \mathrm{negl}(n)$$

This is contradiction for F_k is PRF.

Lemma 3.22 *Π' in Lemma 3.21 is IND-CPA secure.*

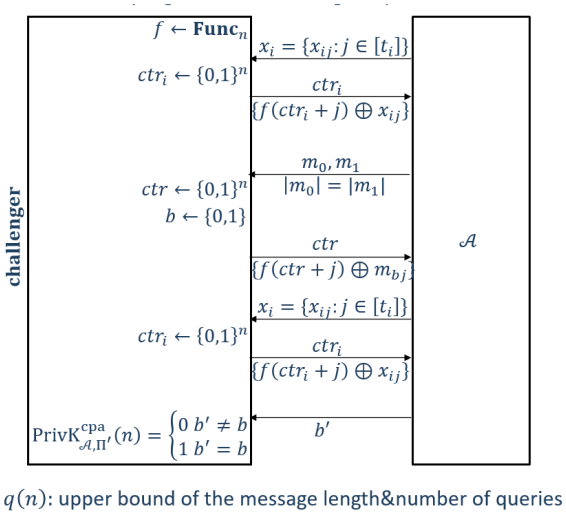


Figure 25: CTR by random function

Proof. Give any adversary $\mathcal{A}$ for Π' in Figure 25. Let I_i be the set $\{\mathrm{ctr}_i + 1, \cdots, \mathrm{ctr}_i + t_i\}$ for plaintext x_i. Let I be the set $\{\mathrm{ctr} + 1, \cdots, \mathrm{ctr} + t_i\}$ for plaintext m_b. Let O_i be the event that $I \cap I' \neq \emptyset$ and O be the event that $\bigvee_{i=1}^{q} O_i$, where q is an upper bound on both messgae length and number of queries.

As $\Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right] = \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1 | O\right] \Pr[O] + \Pr\left[\mathrm{Privk}^{\mathrm{cpa}}_{\mathcal{A},\Pi'}(n) = 1 | \overline{O}\right] \Pr[\overline{O}]$. Consider these two cases:

- If O occurs, then we can clearly determine b, this is

$$\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1|O\right] = 1$$

- If $\overline{O}$ occurs, then each block $f(\text{ctr}+j) \oplus m_{bj}$ are truly random. Since f and ctr are both truly random. We could not determine b. Thus for each b the probability of guess right is $\frac{1}{2}$, this is

$$\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1|\overline{O}\right] = \frac{1}{2}$$

According to above analysis, we have to determine $\Pr[O]$. We have $\Pr[O] \leq \sum_{i=1}^{q} \Pr[O_i]$, so we can consider $\Pr[O_i]$ instead. If $\Pr[O]$ happened, that is two intervals $[\text{ctr}_i + 1, \text{ctr}_i + t_i]$ and $[\text{ctr}, \text{ctr} + t]$ are overlapped. There are two cases:

- $\text{ctr} + 1 \leq \text{ctr}_i + t_i \leq \text{ctr} + t \Rightarrow \text{ctr} + 1 - t_i \leq \text{ctr}_i \leq \text{ctr} + t - t_i$.
- $\text{ctr} + 1 \leq \text{ctr}_i + 1 \leq \text{ctr} + t \Rightarrow \text{ctr} \leq \text{ctr}_i \leq \text{ctr} + t - 1$

Thus we have $\text{ctr} + 1 - t_i \leq \text{ctr}_i \leq \text{ctr} + t - 1$. Once ctr is determined, we have $t + t_i - 1$ choices for ctr_i. Futhermore we have $\Pr[O_i] = \frac{t+t_i-1}{2^n}$ (each ctr has 2^n choices). Therefore we have

$$\Pr[O] \leq \sum_{i=1}^{q} \Pr[O_i] = \sum_{i=1}^{q} \frac{t + t_i - 1}{2^n} \leq q \cdot \frac{2q}{2^n} = \frac{2q^2}{2^n}$$

Finally, we have

$$\Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1\right] \leq \Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1|O\right] \Pr[O] + \Pr\left[\text{Privk}^{\text{cpa}}_{\mathcal{A},\Pi'}(n) = 1|\overline{O}\right] = \frac{2q^2}{2^n} + \frac{1}{2}$$

It is obvious that $\frac{2q^2}{2^n}$ is a negligible function with n.

Theorem 3.23 *If F is a PRF, then CTR is IND-CPA secure.*

Proof. From Lemma 3.21 and 3.22 with triangle inequality(imaging $\frac{1}{2}$ is point).

3.11 Message Authentication Code

Definition 3.35 A message authentication code is a tuple $(\text{Gen}, \text{Mac}, \text{Vrfy})$ of three PPT algorithms, where

- $k \leftarrow \text{Gen}(1^n)$: key generation, $|k| \leq n$.
- $t \leftarrow \text{Mac}(k, m)$: tag generation, $m \in \{0,1\}^*$.
- $\{0,1\} \leftarrow \text{Vrfy}(k, m, t)$: vertification, $\text{Vrfy}(k, m, \text{Mac}(k, m)) = 1$ (correctness).

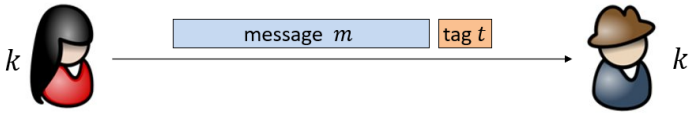


Figure 26: Message authentication code

Suppose adversary wants to construct a (m', t') accepted by Vrfy. So he must get some information about k.

Definition 3.36 A message authentication experiment is Figure 27. Defined as follows.

- Let $\mathcal{A}$ be a adversary.

- do oracle testing.
- output a pair of message and tag (m, t).

- There is a challenger.
  - accept oracle testing
  - verify (m, t) from adversary, output 1 iff $m \in \{m_i\}$ and $t = f(m)$.

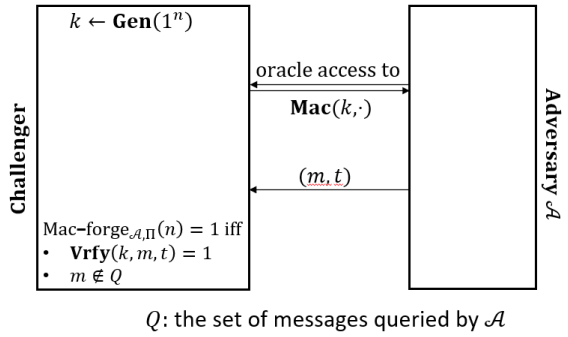


Figure 27: MAC experiment

Definition 3.37 Π existentially unforgeable under an adaptive chosen-message attack (EUF-CMA) if for all PPT adversary $\mathcal{A}$ there is a negligible function negl$(\cdot)$ such that

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

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment.

Note Replay attack: an adversary may intercept (m, t) and send it again. There are two methods to migrate it:

- sequence numbers: need to be synchronized.
- time-stamp: send $T\|m$ instead of m.

Example 3.21 (Fixed-length MAC from PRF) Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a length-preserving PRF. Define $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ as follows.

- $k \leftarrow \text{Gen}(1^n)$: let $k \leftarrow \{0,1\}^n$ and output k.
- $t \leftarrow \text{Mac}(k, m)$: for $m \in \{0,1\}^n$, output $t = F_k(m)$.
- $\{0,1\} \leftarrow \text{Vrfy}(k, m, t)$: output 1 iff $t = F_k(m)$.

Definition 3.38 If MAC verification process is same with MAC construction, then we say it is *canonical verification*.

Lemma 3.24 *Define* $\Pi = (\text{Gen}', \text{Mac}', \text{Vrfy}')$ *as follows.*

- $k \leftarrow \text{Gen}(1^n)$*: let* $f \leftarrow \boldsymbol{Func}_n$ *and output* $k = f$.
- $t \leftarrow \text{Mac}(k, m)$*: for* $m \in \{0,1\}^n$*, output* $t = f(m)$.
- $\{0,1\} \leftarrow \text{Vrfy}(k, m, t)$*: output* 1 *iff* $t = f(m)$.

Then we have

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

Proof. We construct a experiment in Figure 28. $\mathcal{D}$ is a distinguisher for F_k and f, and $\mathcal{A}$ is adversary for both Π and Π'. Consider $\mathcal{O}$ in two cases:

- If $\mathcal{O} = F_k$, then $\mathcal{A}$ is part of Π. We have

$$\Pr\left[\mathcal{D}^{F_k} = 1\right] = \Pr\left[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1\right]$$

- If $\mathcal{O} = f$, then $\mathcal{A}$ is part of Π'. We have

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

Therefore we have

$$\left|\Pr\left[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1\right] - \Pr\left[\text{Mac-forge}_{\mathcal{A},\Pi'}(n) = 1\right]\right| = \left|\Pr\left[\mathcal{D}^{F_k} = 1\right] - \Pr\left[\mathcal{D}^{f} = 1\right]\right| \leq \text{negl}(n)$$

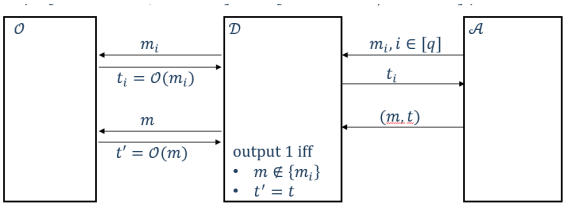


Figure 28: Reduction of MAC by PRF

Lemma 3.25 *The Π' in Example 3.24 is EUF-CMA.*

Proof. The MAC-forge experiment in Figure 29. Consider joint probability $\Pr[m \notin \{m_i\} \wedge t = f(m)] = \Pr\left[\text{Mac-forge}_{\mathcal{A},\Pi'}(n) = 1\right]$, that is

$$\begin{aligned}\Pr[m \notin \{m_i\} \wedge t = f(m)] &= \Pr[m \notin \{m_i\} | t = f(m)] \Pr[t = f(m)] \\ &\leq \Pr[t = f(m)] = \frac{1}{2^n} \qquad (f \text{ is truly random})\end{aligned}$$

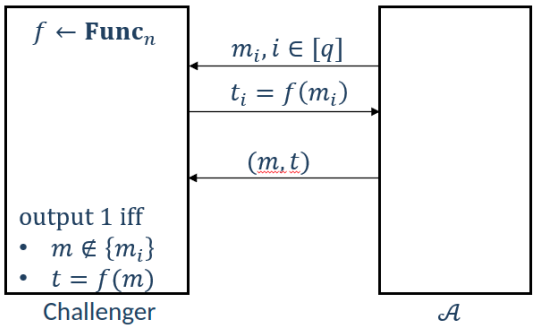


Figure 29: MAC by random function

Theorem 3.26 *If F if a PRF, then Π is EUF-CMA.*

Proof. From Lemma 3.24 and 3.25 with triangle inequality.

How to construct arbitrary-length MAC? Let $m = m_1 \cdots m_d$ and Mac be a fixed-length MAC, some bad ideas:

- affected by block re-order attack: let $t_i = \text{Mac}(k, m_i)$ and output $t_1 \cdots t_d$.

- affected by block truncation attack: let $t_i = \text{Mac}(k, i\|m_i)$ and output $t_1 \cdots t_d$.
- affected by block mix-and-match attack: let $t_i = \text{Mac}(k, l\|i\|m_i)$ (l is length of m) and output $t_1 \cdots t_d$.

Example 3.22 (Arbitrary-length MAC Construction) Let $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ be an arbitrary-length MAC. Define $\Pi' = (\text{Gen}', \text{Mac}', \text{Vrfy}')$ as follows.

- $k \leftarrow \text{Gen}'(1^n) = \text{Gen}(1^n)$.
- $\text{Mac}'(k, m)$: let $|m| = l < 2^{\frac{n}{4}}$, parse m as $m_1, \cdots, m_d$. Each $|m_i| = \frac{n}{4}$ with padding 0.
  - Let $r \leftarrow \{0,1\}^{\frac{n}{4}}$.
  - Let $t_i = \text{Mac}(k, r\|l\|i\|m_i)$ for every $i = 1, \cdots, d$.
  - Let $t = (r, t_1 \cdots t_d)$
- $\text{Vrfy}'(k, m, t)$: Let $t = (r, t_1 \cdots t_{d'})$ and $m = (m_1 \cdots m_d)$. Output 1 iff $d' = d$ and $\text{Vrfy}(k, r\|l\|i\|m_i, t) = t_i$ for each $i = 1, \cdots, d$.

This method is not efficient:

- many times PRF computation: $d \geq \frac{4|m|}{n}$.
- tag is too long: $|t| > 4|m|$.

Note Why $2^{\frac{n}{4}}$? Let the length of $r\|l\|i\|m_i$ be exactly n.

Theorem 3.27 *If Π is EUF-CMA, then Π' in 3.22 is also EUF-CMA.*

Example 3.23 (Fixed-length CBC-MAC) Let F be a length-preserving PRF. Define $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ as follows.

- $\text{Gen}(1^n)$: Let $k \to \{0,1\}^n$.
- $\text{Mac}(k, m)$: Let plain-text $|m| = d \cdot n$, parse m as $m_1 \cdots m_d$, where $|m_i| = n$. Let $t_0 = 0^n$, for every $i > 0$, Let $t_i = F_k(t_{i-1} \oplus m_i)$. Output $t = t_d$.
- $\text{Vrfy}(k, m, t)$: Output 1 iff $|m| = d \cdot n$ and $t = \text{Mac}(k, m)$.

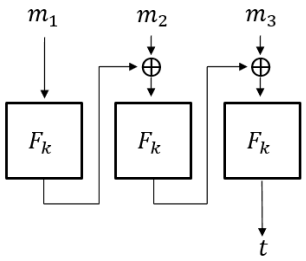


Figure 30: Fixed-length CBC-MAC

Theorem 3.28 *If F is a PRF, then the fixed-length CBC-MAC is an EUF-CMA secure MAC scheme for message of length $d \cdot n$, where d is fixed number once it was chosen.*

Note no length limitation may not EUF-CMA. Given two MAC $t = mac(m_1 \cdots m_d)$, $t' = mac(t \cdots m'_d)$, we can construct a valid MAC $m_1 \cdots m_d \| 0^n \cdots m'_d$, why 0^n? cuz $t \oplus 0^n = t$.

Compare two mac construction:

- Fixed-length CBC-MAC vs. Fixed-length MAC from PRF: CBC-MAC can authenticate $d \cdots n$-bit messages vs. MAC from PRF can only authenticate n-bit message.

- Fixed-length CBC-MAC vs. Arbitrary-length MAC from PRF:
  - CBC-MAC will generate n-bits tags for $d \cdot n$-bit message vs. MAC from PRF will generate $4d \cdot n + \frac{n}{4}$-bit for $d \cdot n$-bit message.
  - CBC-MAC will run PRF d times for $d \cdot n$-bit message vs. MAC from PRF will run PRF $4d$ times for $d \cdot n$-bit message.

Example 3.24 (Arbitrary-length CBC-MAC1) Let CBC-MAC be a MAC function in fixed CBC-MAC. Define $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ as follows.

- $\text{Gen}(1^n)$: Let $k \to \{0,1\}^n$.
- $\text{Mac}(k, m)$: Let m be arbitrary length plain-text. Let $t = \text{CBC-MAC}(k, |m| \| m)$ and output t.
- $\text{Vrfy}(k, m)$: Output 1 iff $t = \text{MAC}(k, |m| \| m)$.

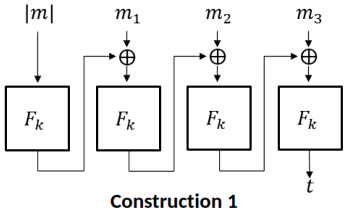


Figure 31: arbitrary-length CBC-MAC1

Example 3.25 (Arbitrary-length CBC-MAC2) Let CBC-MAC be a MAC function in fixed CBC-MAC. Define $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ as follows.

- $\text{Gen}(1^n)$: Let $k_1, k_2 \to \{0,1\}^n$.
- $\text{Mac}(k, m)$: Let m be arbitrary length plain-text. Let $t = \text{CBC-MAC}(k_1, m)$ and $t' = F_{k_2}(t)$, output t'.
- $\text{Vrfy}(k, m)$: Output 1 iff $t = \text{MAC}(k, \| m)$.

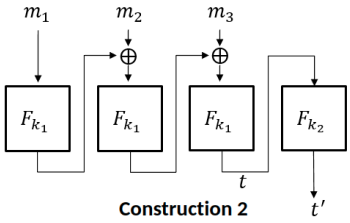


Figure 32: arbitrary-length CBC-MAC2

3.12 Chosen-Cipher-Text Attack

Definition 3.39 The CCA indistinguishability experiment $\text{Privk}^{\text{cca}}_{\mathcal{A},\Pi}(n)$ defined in Figure 33

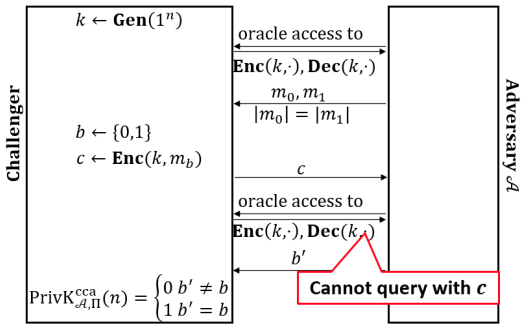


Figure 33: CCA indistinguishability experiment

Note Compare to CCP, the adversary also can send cipher-texts to the oracle, but it doesn't make sense to send the cipher c to the oracle.

Definition 3.40 Π has indistinguishable encryptions under a chosen-ciphertext attack (IND-CCA) if for all PPT adversary $\mathcal{A}$ there is a negligible function $\text{negl}(\cdot)$ such that

$$\Pr\left[\text{Privk}^{\text{cca}}_{\mathcal{A},\Pi}(n)\right] \leq \frac{1}{2} + \text{negl}(n),$$

where the probability if taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment. There is a equivalent notation

$$\left|\Pr\left[\text{Privk}^{\text{cca}}_{\mathcal{A},\Pi}(n)\right] - \frac{1}{2}\right| \leq \text{negl}(n),$$

Remark The encryption schemes constructed so far are not IND-CCA, both are affected by malleability which easily transform a valid cipher to another valid one.

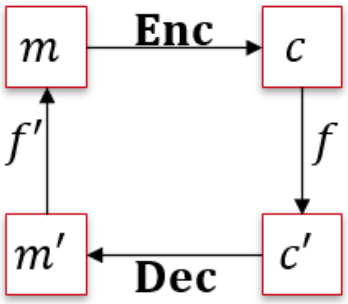


Figure 34: malleability

Example 3.26 Let Π be OTP: $\text{Enc}(k, m) = k \oplus m$. We can construct an adversary as follows.

- Let $m_0 = 0^n, m_1 = 1^n$, and let challenger output $c = \text{Enc}(k, m_b) = k \oplus m_b = x_1 x_2 \cdots x_n$.
- Query the oracle with cipher $c' = (x_1 \oplus 1)x_2 \cdots x_n$, and let oracle output m' for c'.
- If $m' = 01^{n-1}$ (that is $m_1 \oplus 1$), output $b' = 1$, otherwise if $m' = 10^{n-1}$ (that is $m_0 \oplus 1$), output $b' = 0$.

Example 3.27 For models of operation:

- OFB: Flip one bit of c_1 and query the decryption oracle, do same action like in 3.26.

- CTR: same like OFB.
- ECB: Let $m_0 = 0^n1^n, m_1 = 1^n0^n$, and let challenger output $c = c_1c_2$. Query the decryption oracle with the cipher $c' = c_2c_1$ and get m' from the oracle. If $m' = 0^n1^n$ outputs 1, otherwise output 0.
- CBC: Let $m_0 = 0^n, m_1 = 1^n$, and let challenger output $c = (IV, x)$. Query the decryption oracle with the cipher $c' = (IV', x)$, get m' from oracle, where $IV' = IV$ except the 1st bit. If $m' = 01^{n-1}$, output $b' = 1$, otherwise output $b' = 0$.

Definition 3.41 The strong message Authentication experiment MAC-sforge$_{\mathcal{A},\Pi}(n)$ is defined in Figure 35.

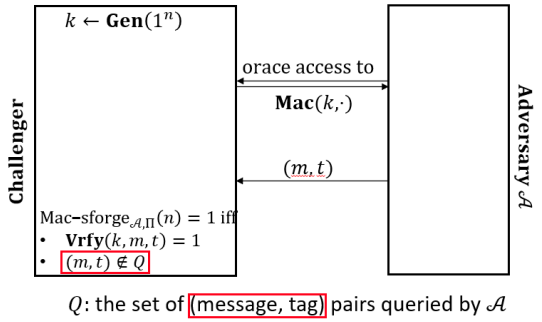


Figure 35: Strong message Authentication experiment

Note it requires weaker verification in Vrfy(k, m, t) compared to Mac-forge. That is if we can prove safety in some weaker verification, then it must be safe in stronger verification.

Definition 3.42 Π is strongly existentially unforgeable under an adaptive chosen-message attack (sEUF-CMA) strongly secure if for all PPT adversary $\mathcal{A}$ there is a negligible function negl$(\cdot)$ such that

$$\Pr\left[\text{MAC-sforge}_{\mathcal{A},\Pi}(n) = 1\right] \leq \text{negl}(n),$$

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment.

Example 3.28 (IND-CCA scheme) Let $\Pi_E = (\text{Gen}_E, \text{Enc}_E, \text{Dec}_E)$ be an IND-CPA secure encryption. Let $\Pi_M = (\text{Gen}_M, \text{Mac}, \text{Vrfy})$ be a sEUF-CMA secure MAC. Define $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ as follows.

- Gen(1^n): let $k_E \leftarrow \text{Gen}_E$, $k_M \leftarrow \text{Gen}_M(1^n)$, output $k = (k_E, k_M)$.
- Enc(k, m): let $c = \text{Enc}_E(k_E, m)$, $t = \text{Mac}(k_M, c)$, output (c, t).
- Dec$(k, (c, t))$: If Vrfy$(k_m, c, t) = 0$, output $\perp$, otherwise output $m = \text{Dec}_E(k_E, c)$.

Note sEUF-CMA secure MAC in 3.28 will make $c \xrightarrow{f} c'$ hard, this is attacker can only use the cipher of (m, c) that used for oracle. Thus the encryption oracle will be barely useful for attacker. The remind encryption will be a CPA scheme, that is Π_E.

3.13 Hash Function

Definition 3.43 A hash function is a pair $\Pi = (\text{Gen}, H)$ of PPT algorithms defined as follows.

- Let $s \leftarrow \text{Gen}(1^n)$ be public key.
- Let $H^s : \{0,1\}^* \to \{0,1\}^{l(n)}$ be hash function from arbitrary-length to fixed-length.

Note A fixed-length hash function is only accepted fixed-length inputs.

Definition 3.44 The collision-finding experiment is Hash-coll$_{\mathcal{A},\Pi}(n)$ defined in Figure 36.

- Challenger give out the hash algorithm.
- Adversary need find two plain-text x, x' such that $x \neq x'$ and $H^s(x) = H^s(x')$.

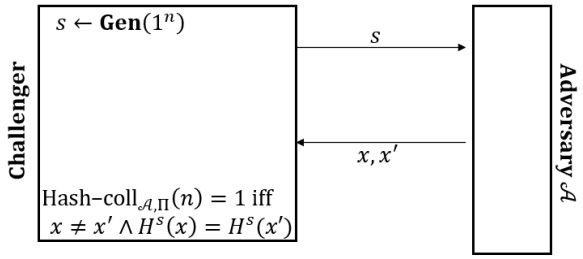


Figure 36: Collision-finding experiment

Definition 3.45 Π is collision resistance if for all PPT adversary $\mathcal{A}$ there is a negligible function $negl(\cdot)$ such that

$$\Pr\left[\text{Hash-coll}_{\mathcal{A},\Pi}(n) = 1\right] \leq \text{negl}(n).$$

Definition 3.46 The second-preimage finding experiment is Hash-SecPre$_{\mathcal{A},\Pi}(n)$ defined in Figure 37.

- Challenger give out the hash algorithm and a plain-text x.
- Adversary need find two plain-text x' such that $x \neq x'$ and $H^s(x) = H^s(x')$.

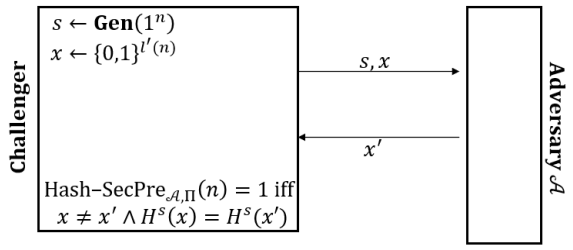


Figure 37: Second-preimage finding experiment

Definition 3.47 Π is second-preimage resistance if for all PPT adversary $\mathcal{A}$ there is a negligible function negl$(\cdot)$ such that

$$\Pr\left[\text{Hash-SecPre}_{\mathcal{A},\Pi}(n) = 1\right] \leq \text{negl}(n).$$

where the probability is taken over all random coins in the experiment.

Definition 3.48 The preimage finding experiment is Hash-Pre$_{\mathcal{A},\Pi}(n)$ defined in Figure 38.

- Challenger give out the hash algorithm and a hash value y.
- Adversary need find two plain-text x such that $H^s(x) = y$.

Definition 3.49 Π is preimage resistance if for all PPT adversary $\mathcal{A}$ there is a negligible function negl$(\cdot)$ such that

$$\Pr\left[\text{Hash-Pre}_{\mathcal{A},\Pi}(n) = 1\right] \leq \text{negl}(n).$$

where the probability is taken over all random coins in the experiment.

Remark Collision resistance $\Rightarrow$ Second-preimage resistance $\Rightarrow$ Preimage resistance.

We construct arbitrary-length hash function from fixed-length hash

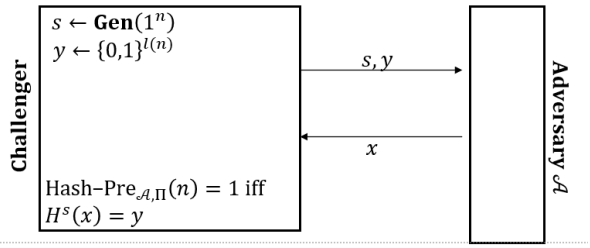


Figure 38: Preimage finding experiment

Definition 3.50 (Merkle-Damgård Transform) Let $\Gamma = (\text{Gen}, h) : \{0,1\}^{2n} \to \{0,1\}^n$ be a fixed-length hash. Define arbitrary-length hash function $\Pi = (\text{Gen}', H) : \{0,1\}^* \to \{0,1\}^n$ as follows.

- $\text{Gen}' : s \leftarrow \text{Gen}(1^n)$.
- $H^s(x)$: Let $x \in 0, 1^*$ and $|x| = L < 2^n$ (L could be encoded in n bits), then
  - Let $B = \lceil \frac{l}{n} \rceil$, and may pad x with 0 to $x_1 \cdots x_B$ such that $|x_i| = n$ for every i.
  - Let $x_{B+1} = [L]$, where $[L] \in \{0,1\}^n$ is bits representation of L.
  - Let $z_0 = 0^n$ (z_0 could be arbitrary value with fixed-length n).
  - For each $i = 1, \cdots, B+1$, let $z_i = h^s(z_{i-1} \| x_i)$.
  - Output z_{B+1}.

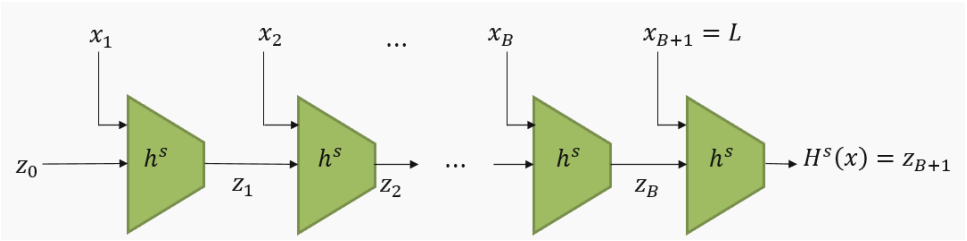


Figure 39: (Merkle-Damgård Transform

Theorem 3.29 *If Γ is collision resistant, then Π is also collision resistant.*

Proof. Suppose Π is not collision resistan. Then there is a PPT algorithm $\mathcal{A}$ such that

$$\Pr\left[\text{Hash-coll}_{\mathcal{A},\Pi}(n) = 1\right] = \epsilon.$$

is non-negligible in security parameter n. Assume (x, x') $(x \neq x')$ is a collision $(H^s(x) = H^s(x'))$ for H^s. Consider hash process of (x, x') in Figure 40. In the last hash function h^s call, there are two cases for inputs:

- $x_{B+1} \| z_B = x'_{B'+1} \| z'_{B'}$: that is $x_{B+1} = x_{B'+1}$ and $z_B = z'_{B'}$.
- $x_{B+1} \| z_B \neq x'_{B'+1} \| z'_{B'}$: we find a collision $(x_{B+1} \| z_B, x'_{B'+1} \| z'_{B'})$ for Γ.

For the case1, we can keep considering the second-last hash function h call use above method and go on. Since B and B' is finite, it will be terminated in some where. The only case that we can't find a collision pair for Γ is that all hash function h^s calls had been considered. That is $x_1 || z_0 = x'_1 || z'_0$, is contradictory to $x \neq x'$. Thus we can always find a collision pair (y, y') for Γ from (x, x'). That is

$$\Pr\left[\text{Hash-coll}_{\mathcal{A},\Gamma}(n) = 1\right] \geq \Pr\left[\text{Hash-coll}_{\mathcal{A},\Pi}(n) = 1\right] = \epsilon,$$

is a contradictory.

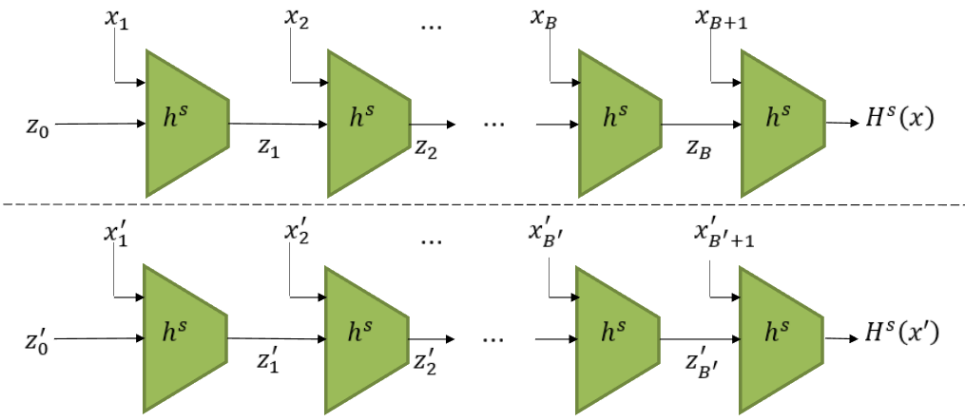


Figure 40: (Merkle-Damgård Transform for (x, x_i)

Example 3.29 (Hash and MAC) Let $\Pi = (\text{Gen}, \text{MAC}, \text{Vrfy})$ be a fixed-length MAC for $l(n)$-bit messages, and $\Pi_H = (\text{Gen}_H, H) : \{0,1\}^* \to \{0,1\}^{l(n)}$ be a hash function with output length $l(n)$ bits. Define an arbitrary-length MAC $\Pi = (\text{Gen}', \text{MAC}', \text{Vrfy}')$ as follows.

- $\text{Gen}'(1^n)$: let $k \leftarrow \text{Gen}(1^n)$ (private) and $s \leftarrow \text{Gen}_H(1^n)$ (public), output $k' = (k, s)$.
- $\text{Mac}'(k', m)$: let $t = \text{Mac}(k, H^s(m))$, output t.
- $\text{Vrfy}'(k', m, t)$: output 1 iff $\text{Vrfy}(k, H^s(m), t) = 1$.

Theorem 3.30 *In Ex 3.29. If Π is EUF-CMA and Π_H is collision-resistant, then Π' is EUF-CMA.*

Example 3.30 (HMAC) Let (Gen_H, H) be a MD transform of fixed-length hash function (Gen_h, h). Define a arbitrary-length MAC $\Pi = (\text{Gen}, \text{MAC}, \text{Vrfy})$ as follows.

- $\text{Gen}(1^{512})$: let $k \leftarrow \{0,1\}^{512}$.
- $\text{Mac}(k, m)$: let

$$t = H^s(h^s(k \oplus opad) \| H^s(k \oplus ipad \| m)),$$

  output t.

where $opad = \underbrace{0\text{x}5c \cdots 5c}_{512 \text{ bits}}$ and $ipad = \underbrace{0\text{x}37 \cdots 37}_{512 \text{ bits}}$.

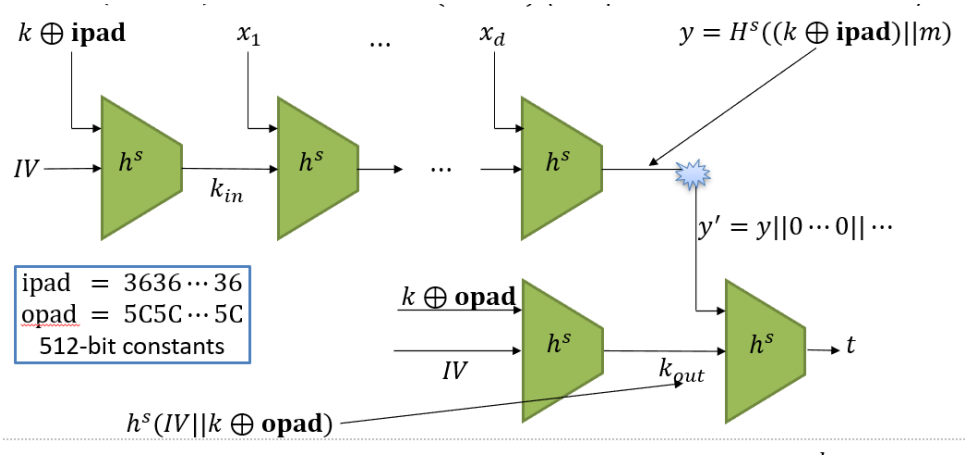


Figure 41: HMAC construction

Given a hash function $H : \{0,1\}^* \to \{0,1\}^l$ with fixed length digest. How to find a collision for H in general? There are two attack:

- Trivial attack: choose $2^l + 1$ distinct message, then there must exist two messages with same hash.
- Birthday attack: choose $O(2^{\frac{l}{2}})$ distinct message, check whether there exist two messages with same hash.

Note Birthday attack does not guarantee the finding of collision. If the probability of successful attack in birthday attack is not bad, then we can use the birthday attack multiple times. Since $O(2^{l/2}) \ll 2^l$.

Given a set of distinct messages $x_1, x_2, \cdots, x_q$, the probability of hash collision is

$$\Pr\left[\exists i, j \in [q].\ i \neq j \wedge H(x_i) = H(x_j)\right]$$

If we treat H as a truly random function. Let $y_1, \cdots, y_q \leftarrow \{0,1\}^l$, the above probability is equivalent to

$$\Pr\left[\exists i, j \in [q].\ i \neq j \wedge y_i = y_j\right]$$

we use $\mathrm{Coll}(q, N)$ as above probability where N is symbol for the cardinality of selected set. i.e., 2^l for $\{0,1\}^l$.

Lemma 3.31 $\mathrm{Coll}(q, N) \leq \frac{q^2}{2N}$.

Proof. Let $C_{i,j}$ be the event that $y_i = y_j$, where $i \neq j$. Then we have

$$\mathrm{Coll}(q, n) = \Pr\left[\bigvee_{i \neq j} C_{i,j}\right] \leq \sum_{i \neq j} \Pr\left[C_{i,j}\right] = \binom{q}{2} \cdot \frac{1}{N} = \frac{q(q-1)}{2N} \leq \frac{q^2}{2N}$$

Lemma 3.32 *If* $q \leq \sqrt{2N}$, *then* $\mathrm{Coll}(q, N) \geq 1 - e^{-\frac{q(q-1)}{2N}} \geq \frac{q(q-1)}{4N}$.

Proof. Let C be the event $\bigvee_{i \neq j} C_{i,j}$, then $\neq C$ is the event that there is no collision in $y_1, \cdots, y_q$, that is $y_1, \cdots, y_q$ are distinct. Then we have

$$\mathrm{Coll}(q, n) = 1 - \Pr\left[\neg C\right] = 1 - \binom{N}{q} = \prod_{i}^{q-1}(1 - \frac{i}{N})$$

According to $1 - x \leq e^{-x}$ in $[0, 1]$, we have $1 - \frac{i}{N} \leq e^{-\frac{i}{N}}$ for every $i = 1, \cdots, q-1$. Then we have

$$\mathrm{Coll}(q, n) \geq e^{-\sum_i^{q-1} \frac{i}{N}} = e^{-\frac{q(q-1)}{2N}}.$$

According to $e^{-x} \leq 1 - \frac{x}{2}$ in $[0, 1]$.When $q \leq \sqrt{2N}$, the $\frac{q(q-1)}{2N} \leq \frac{2N - \sqrt{2N}}{2N} = 1 - \frac{1}{\sqrt{2N}} < 1$. Thus we have

$$\mathrm{Coll}(q, n) \geq \frac{q(q-1)}{4N}.$$

Thus

Remark In Lemma 3.32, $\sqrt{2N}$ is not tight upper bound. Since the inequality $e^{-x} < 1 - \frac{x}{2}$ is also preserved in a bit large q when $q > 1$ with proper N, such that $x = 1 + \frac{1}{\sqrt{4}}$ i.e., for $q = \sqrt{2N} + 1$, we have $\frac{q(q-1)}{2N} = 1 + \frac{1}{\sqrt{2N}}$, the N should be large than 2.

If we want $\mathrm{Coll}(q, N) \geq \frac{1}{2}$, that is $\frac{q(q-1)}{4N} \geq \frac{1}{2}$, we can get the least $q = \sqrt{2N} + 1$. When we let $N = \frac{2}{l}$, we can get least $q = \sqrt{2} \cdot 2^{\frac{l}{2}} + 1$ in the same way. That is why n-bit security requires digest length $l \geq 2n$. We have regarded the hash function H as truly random, when the H is not truly random, attacker may use less time. If adversary can run in time T maximum, then we must require $T < \frac{2}{\frac{2}{l}}$, that is $l > 2 \log T$.

3.14 Data Encryption Standard (DES)

DES is a practical block cipher construction. Block ciphers require PRPs or sPRPs $F : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^l$, where n is called the key length and l is called the block length.

Example 3.31 The FIPS 46-3 DES is in Figure 42. There are annotations for this scheme as follows.

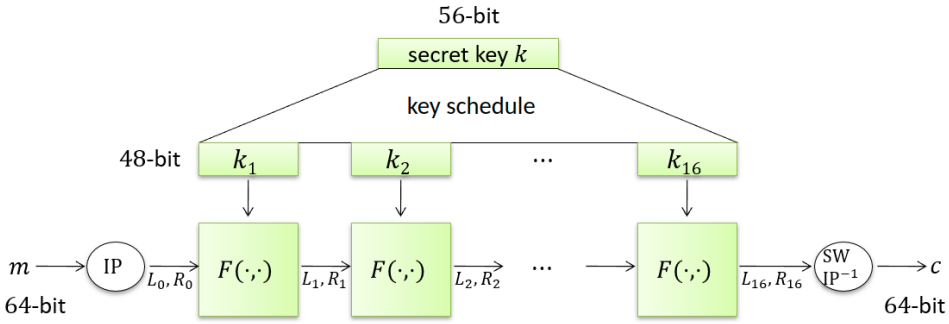


Figure 42: (FIPS 46-3 DES

- The key length $n = 56$ bits (actually 64, 56 is valid number); the block length is $m = 64$.
- The L_i and R_i are both 32-bits, and calculated by Feistel network.

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(k_i, R_{i-1})$$

In inverse:

$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus F(k_i, L_i)$$

- The IP is called initial permutation, it is a table in Figure 43a. The inverse IP is a table in Figure 43b.

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

- $y_1 = x_{58}$
- $y_2 = x_{50}$
- $y_3 = x_{42}$
- $y_4 = x_{34}$
- $y_5 = x_{26}$
- $y_6 = x_{18}$
- $y_7 = x_{10}$
- $y_8 = x_2$
- ...
- $y_{64} = x_7$

(a) IP

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

- $x_1 = y_{40}$
- $x_2 = y_8$
- $x_3 = y_{48}$
- $x_4 = y_{16}$
- $x_5 = y_{56}$
- $x_6 = y_{24}$
- $x_7 = y_{64}$
- $x_8 = y_{32}$
- ...
- $x_{64} = y_{25}$

(b) IP^{-1}

Figure 43: initial permutation IP

- The $F : \{0,1\}^{48} \times \{0,1\}^{64} \to \{0,1\}^{64}$ is a PRP (sPRP). Define as

$$F(k_i, L_{i-1}, R_{i-1}) = (R_{i-1}, L_{i-1} \oplus \hat{f}(k_i, R_{i-1})) = (L_i, R_i)$$

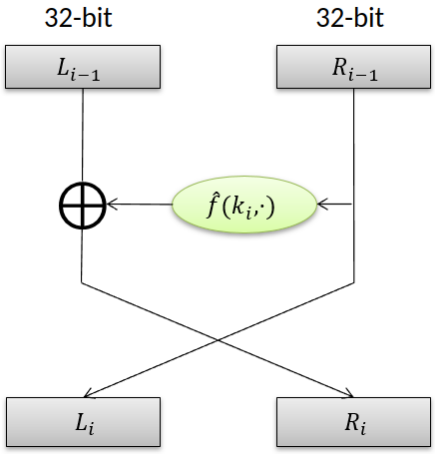
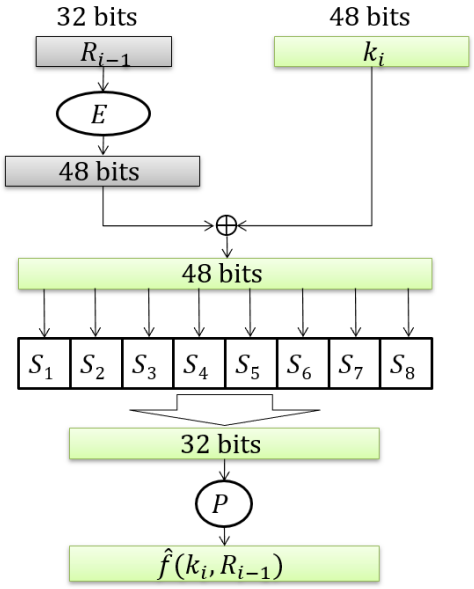


Figure 44: One feistel network $F(\cdot,\cdot)$

where $\hat{f} : \{0,1\}^{48} \times \{0,1\}^{32} \to \{0,1\}^{32}$ is DES mangler function in Figure 45a. The E is expansion function for 32-bits message to 48-bits in 45b. The last P is a permutation in 45c. The $S_i : \{0,1\}^6 \to \{0,1\}^4$ are amazing boxes, i.e. S_3 is in Figure 45d, it is defined as follows.

- $S_3(a_5a_4a_3a_2a_1a_0) = \theta(a_5a_0, a_4a_3a_2a_1)$, it means take two bits a_5a_0 as the number of row i and $a_4a_3a_2a_1$ as the number of column j. Then we take the (i,j) element in Figure 45d as a output. Note the number in 45d are less than 16 such that could always be encoded in 4-bits.



(a) The DES mangler function $F(k_i,\cdot)$

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

(b) Expansion function E

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

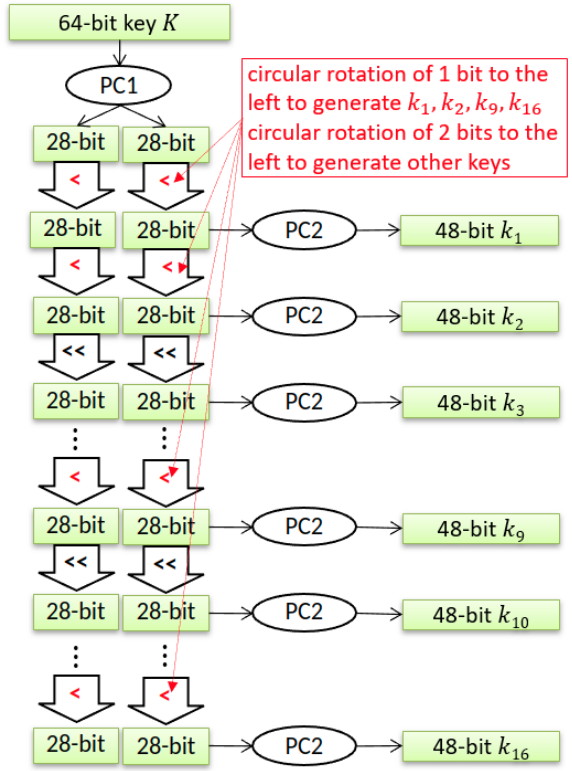
(c) P

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

(d) S_3

Figure 45: S_3 box

- Use the secret key generates 16 keys for F in Figure 46. The key schedule is in Figure 46a, the $<$ is left shift operation and $\ll$ is double left shift operation. Note k_1, k_2, k_9, k_{16} is generated by $<$, and otherwise are $\ll$. PC_1 is a transform for 64-bits key to 56-bits, PC_2 is a transform for 56-bits key to 48-bits.



(a) Key Schedule

PC1

Left							Right						
57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	26	18	7	62	54	46	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

64-bit $K \rightarrow$ 56-bit k

(b) PC_1

PC2

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

56-bit→48-bit

(c) PC_2

Figure 46: Initial permutation IP

- Unfortunately, this encryption is not safe (.

Example 3.32 (Double DES) Define $E_2((k_1, k_2), m) = \text{DES}(k_2, \text{DES}(k_1, m))$. The key length of E_2 is $2n = 112$. But it is not secure necessarily. We could construct a meet-in-the-middle-attack by the fact

$$\text{DES}(k_1, m) = \text{DES}^{-1}(k_2, c)$$

for any given plain-cipher pair (m, c). That says we can construct two set $L_1 = \{(\text{DES}(k_1, m), k_1) | k_1 \in \{0,1\}^n\}$ and $L_2 = \{(\text{DES}^{-1}(k_2, c), k_2) | k_2 \in \{0,1\}^n\}$. Then we can get derive $S = \{(k_1, k_2) | D\}\text{DES}(k_1, m) = \text{DES}^{-1}(k_2, c)$ by comparing L_1 and L_2. Suppose the PRP $F : \{0,1\}^n \times \{0,1\}^l \rightarrow \{0,1\}^l$ in DES is truly random such that F_k is truly random in **Func**. Then $\Pr[s \in L_1 \wedge s \in L_2] = \frac{1}{2^l}$, thus $|S| \approx \frac{2^n \times 2^n}{2^l} = 2^{2n-l}$. If $|S|$ is large

in once test, we can take another pair (m', c') to do same work, to reduce $|S|$, i.e., 2-3 times is enough in practice. The complexity of time for the attack is consist of:

- We could order L_1 and L_2 for comparing in linear time. The ordering method could be
  - counting: $O(2^n)$, it needs $(l+n) \cdot 2^n$ space.
  - general: $O(2^n \cdot \log 2^n) = O(n \cdot 2^n)$, it needs $O(2^n)$ space.
- We need test each $s \in S$ to check right keys, this need time $O(2^{2n-l})$.

Example 3.33 (Triple DES) There are three construction:

- $E_3((k_1, k_2, k2), m) = \text{DES}(k_3, \text{DES}^{-1}(k_2, \text{DES}(k_1, m)))$ with key length 168. Could be attacked in 2^112.
- $E_3((k_1, k_2), m) = \text{DES}(k_1, \text{DES}^{-1}(k_2, \text{DES}(k_1, m)))$ with key length 112. Could be attacked in 2^112.
- $E_3((k_1, k_2, k_3), m) = k_1 \oplus \text{DES}(k_2, k_3 \oplus m)$ with key length 184. Could be attacked in 2^120.

3.15 Advanced Encryption Standard

AES is based on block cipher with (s)PRG $F : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^l$, where $n \in \{128, 192, 256\}$ and $l = 128$.

Example 3.34 (128-AES) The 128-bits key AES encryption in Figure 47. It consists of

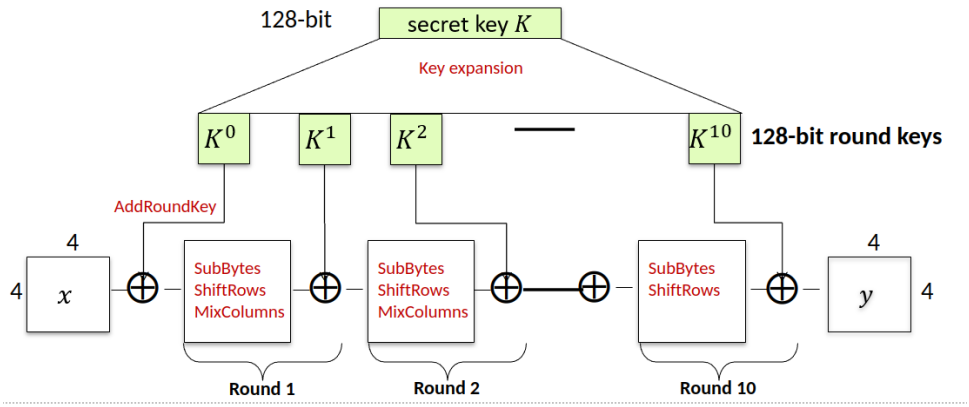


Figure 47: 128 AES encryption

- plaintext/ciphertext are both matrix of 16 bytes, called state.
- key expansion: 128-bit secret key to round keys (128-bits).
- Round 1-Round 9: SubBytes, ShiftRows, MixColumns, AddRoundKey.
- Round 10: SubBytes, ShiftRows, AddRoundkey.

A state in DES is defined in Figure 48. The data is ordered by from top to bottom and left to right, that is $S = S_{0,0}S_{1,0}S_{2,0}S_{3,0}S_{0,1}S_{1,1}S_{2,1}S_{3,1}S_{0,2}S_{1,2}S_{2,2}S_{3,2}S_{0,3}S_{1,3}S_{2,3}S_{3,3}$.

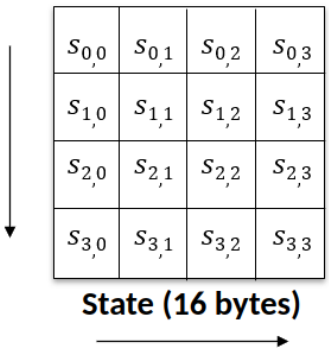


Figure 48: AES state

A key expansion is defined in Figure 49. The secret key is also encoded with state matrix. The w_i is abbr., word. The round-key K^i is generated as follows.

- Let $w_0 = k_{0,0}k_{1,0}k_{2,0}k_{3,0}, w_1 = k_{0,1}k_{1,1}k_{2,1}k_{3,1}, w_2 = k_{0,2}k_{1,2}k_{2,2}k_{3,2}, w_3 = k_{0,3}k_{1,3}k_{2,3}k_{3,3}$. Otherwise
- If $i \neq 0 \mod 4$, let $w_i = w_{i-1} \oplus w_{i-4}$.
- If $i = 0 \mod 4$, let $w_i = \text{SubWord}(\text{RotWord}(w_{i-1})) \oplus RCon[i/4] \oplus w_{i-4}$.
  - RotWord$(A, B, C, D) = (B, C, D, A)$, where A, B, C, D are bytes.

- SubWord$(A, B, C, D) = (S(A), S(B), S(C), S(D))$, where A, B, C, D are bytes and $S(A), S(B), S(C), S(D)$ is in S-Box.

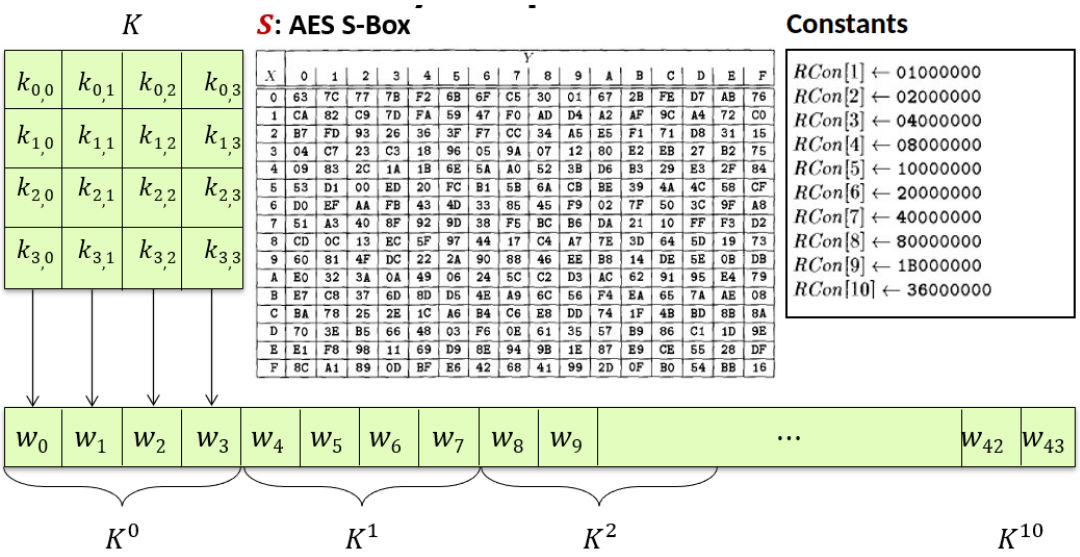


Figure 49: AES key expansion

There are operations for each round encryption.

- AddRoundKey(s, K^i): XOR the state s with the round key K^i in Figure 50. That is $S_{i,j} = s_{i,j} \oplus k_{i,j}$.

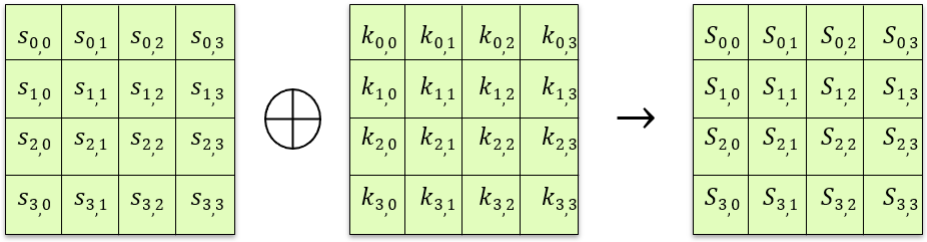


Figure 50: AES AddRoundKey

- SubBytes(s): Substitute the state s with corresponded value in S-Box,that is $S_{i,j} = S(X, Y)$ for each $s_{i,j} = (X, Y)$.
- ShiftRows(s): Shift the rows of state to the left in Figure 51. That is $S_{i,j} = s_{i,j+1}$.

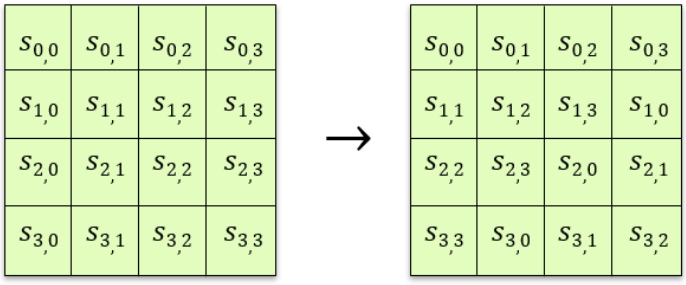


Figure 51: AES ShiftRows

- MixColumns(s): Multiplication over $\mathbb{F}_{2^8}$ in Figure 52. That $\mathbb{F}_{2^8} = \mathbb{Z}_2[X]/(X^8 + X^4 + X^3 + X + 1) = \{\sum_{i=0}^{7} c_i X^i \mid c_0, \cdots, c_7 \in \mathbb{Z}_2\}$. The 1th row of left matrix in Figure 52 is $(X, X + 1, 1, 1)$ (one bit

correspond one coefficient, i.e. highest bit in byte correspond the coefficient X^7). Then $S_{i,j}$ can be calculated in $\mathbb{F}_{2^8}$ and back the result in one byte.

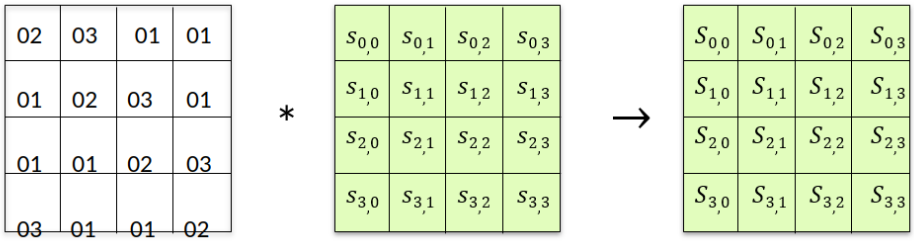


Figure 52: AES MixColumns

AES is secure against all known attacks.

3.16 Discrete Logarithm

Definition 3.51 Let $\mathcal{G}$ be a cyclic group generator, it accept the security parameter 1^n, and output (q, G, g) where $|q| = n$ and $G = \langle g \rangle$ is a cyclic group of order q.

Definition 3.52 Let $G = \langle g \rangle$ be a cyclic group of order q with generator g. For every $h \in G$, there exists $x \in \{0, 1, \cdots, q-1\}$ such that $h = g^x$. The integer x is called the discrete logarithm of h with respect to g, written $x = \log_g h$.

Definition 3.53 The Discrete logarithm experiment $\text{DLog}_{A,\mathcal{G}}(n)$ is defined in Figure 53.

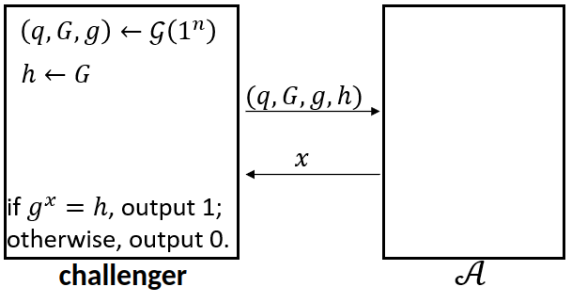


Figure 53: Discrete Logarithm Experiment

Definition 3.54 We say that the discrete logarithm problem is hard relative to $\mathcal{G}$ if for all PPT algorithms $\mathcal{A}$, there is a negligible function negl such that

$$\Pr\left[\text{DLog}_{A,\mathcal{G}}(n) = 1\right] \leq \text{negl}(n).$$

Note The best known algorithm $\exp(O((\ln q)^{1/3}(\ln \ln q)^{2/3}))$.

Definition 3.55 The CDH (Computational Diffie-Hellman) experiment $\text{CDH}_{\mathcal{A},\mathcal{G}}(n)$ is defined in Figure 54.

Definition 3.56 We say the CDH problem is hard relative to $\mathcal{G}$ if for all PPT algorithms $\mathcal{A}$, there is a negligible function negl such that

$$\Pr\left[\text{CDH}_{\mathcal{A},\mathcal{G}}(n)\right] \leq \text{negl}(n).$$

Note The best known algorithm is via solving DLOG first.

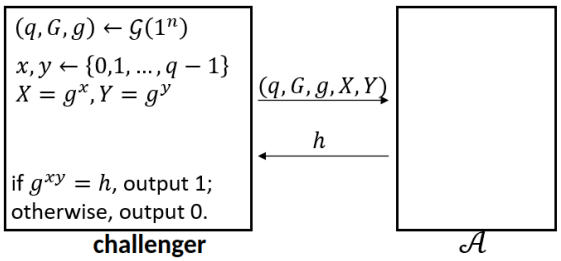


Figure 54: Computational Diffie-Hellman Experiment

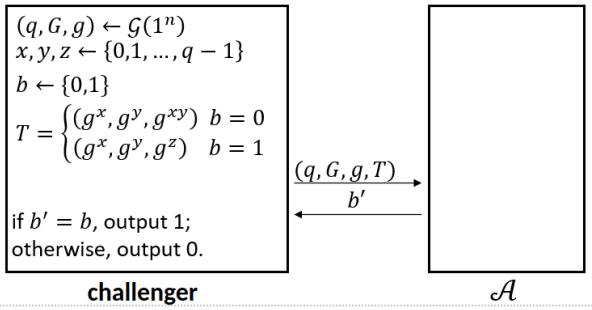


Figure 55: Decisional Diffie-Hellman Experiment

Definition 3.57 The DDH (Decisional Diffie-Hellman) experiment $\text{DDH}_{\mathcal{A},\mathcal{G}}(n)$ is defined in Figure 55.

Definition 3.58 We say the DDH problem is hard relative to $\mathcal{G}$ if for all PPT algorithms $\mathcal{A}$, there is a negligible function negl such that

$$|\Pr[\mathcal{A}(q, G, g, g^x, g^y, g^z) = 1] - \mathcal{A}(q, G, g, g^x, g^y, g^{xy}) = 1| \leq \text{negl}(n).$$

where $x, y, z \leftarrow \{0, 1, \cdots, q-1\}$.

3.17 Key Exchange

There are two main drawbacks of private-key scheme:

- key distribution.
- key storage.

Definition 3.59 A key exchange protocol is a pair of interactive (probability polynomial-time) algorithms $\Pi = (\text{Alice}, \text{Bob})$. It follows below rules.

- The algorithms start with the security parameter n.
- The algorithms compute and send message to each other.
- At the end, Alice outputs $k_A \in \{0,1\}^n$; Bob outputs $k_B \in \{0,1\}^n$ satisfies two goals
  - correctness: $k_A = k_b$;
  - security: any other observer of the communication cannot learn k_A and k_B.

Definition 3.60 The Key exchange experiment is defined in Figure 56.

Definition 3.61 The key exchange protocol Π is secure in the presence of an eavesdropper for all PPT adversary $\mathcal{A}$ there is a negligible function $\text{negl}(\cdot)$ such that

$$\Pr\left[\text{KE}^{\text{eav}}_{\mathcal{A},\Pi} = 1\right] \leq \frac{1}{2} + \text{negl}(n),$$

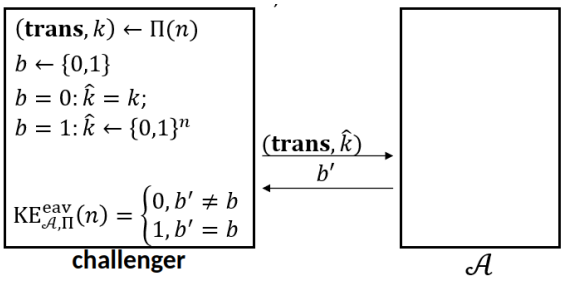


Figure 56: Key Exchange Experiment

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment. There is an equivalent definition

$$\left|\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}=1\right]-\frac{1}{2}\right|\leq \mathrm{negl}(n)$$

Example 3.35 (Diffie-Hellman key exchange) Let $(q,G,g)\leftarrow\mathcal{G}(1^n)$ be a cyclic group generator. We could construct a key exchange protocol as follows.

- Alice side:
  - $(q,G,g)\leftarrow\mathcal{G}(1^n)$.
  - $x\leftarrow\mathbb{Z}_q, h_A=g^x$.
  - send (q,G,g,h_A) to Bob.
- Bob side:
  - $y\leftarrow\mathbb{Z}_q, h_B=g^y$;
  - send h_B to Alice;
  - output $k_B=h_A^y$.
- finally Alice output $k_A=h_B^x$.

Its correctness can be checked from

$$h_A^y=g^{xy}=h_B^x.$$

Theorem 3.33 *Π in Example 3.35 is secure assuming that DDH problem is hard relative to $\mathcal{G}$.*

Proof. Suppose Π is not secure. Then there exists an adversary $\mathcal{A}$ such that

$$\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=1\right]=\frac{1}{2}+\epsilon$$

for some non-negligible function ϵ. We show it is a contradiction by constructing an alogorithm $\mathcal{B}$ that can solve the DDH problem. We present $\mathcal{B}$ combined with $\mathcal{A}$ in Figure 57. $\mathcal{B}$ give $\mathcal{A}$ the trans scripts and h as the key; $\mathcal{A}$'s

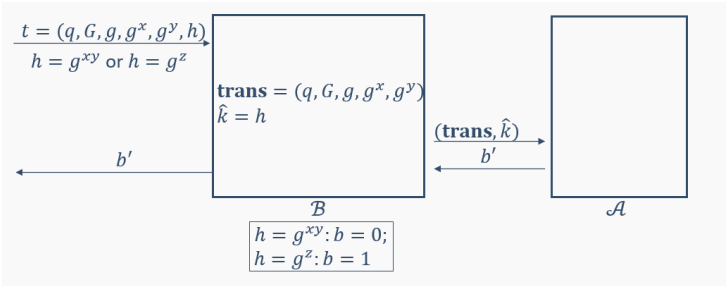


Figure 57: Counterexample algorithm to solve DDH problem

output as $\mathcal{B}$'s output. We consider the relative probability

$$\Pr\left[\mathcal{B}(q,G,g,g^x,g^y,g^{xy})=1\right]=\Pr\left[b'=1|b=0\right]=\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=0|b=0\right]$$
$$\Pr\left[\mathcal{B}(q,G,g,g^x,g^y,g^z)=1\right]=\Pr\left[b'=1|b=1\right]=\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=1|b=1\right]$$

Then we have

$$\begin{aligned}
&\left|\Pr\left[\mathcal{B}(q,G,g,g^x,g^y,g^z)=1\right]-pr\mathcal{B}(q,G,g,g^x,g^y,g^{xy})=1\right|\\
&=\left|\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=1|b=1\right]-\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=0|b=0\right]\right|\\
&=\left|\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=1|b=1\right]-\left(1-\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=1|b=0\right]\right)\right|\\
&=\left|2(\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)=1\right])-1\right|\\
&=|2\epsilon|
\end{aligned}$$

This is a contradiction.

Diffie-Hellman key exchanger suffers the main-in-the-middle attack, Figure 58 show how it works.

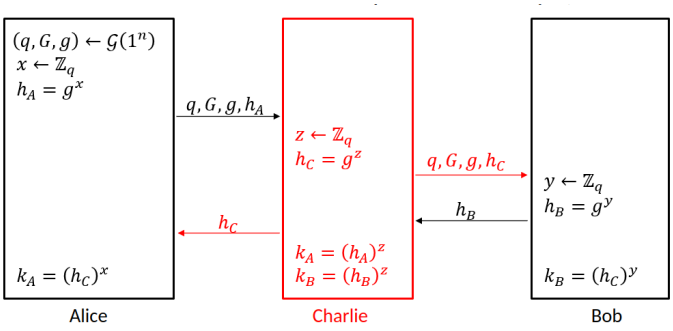


Figure 58: Counterexample algorithm to solve DDH problem

4 Public-Key Encryption

4.1 Introduction

A Public-key encryption consists with three component: key generation **Gen**, encryption **Enc** and decryption **Dec**. Key generation Gen have to give two keys: public key pk and private key sk. Given a plaintext m, the corresponding cipher $c = \text{Enc}(pk, m)$. Correctness is guaranteed by

$$\text{Dec}(sk, \text{Enc}(pk, m)) = m$$

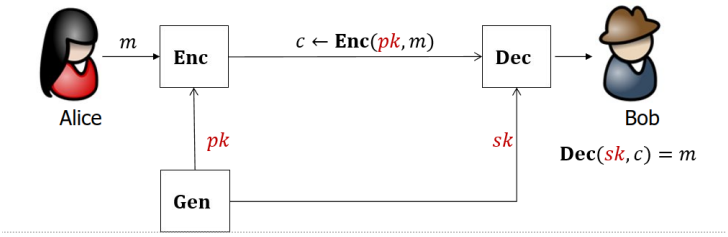


Figure 59: Public-Key Encryption

Definition 4.1 Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ be a public-key encryption. Define an adversarial indistinguishability experiment $\text{PubK}^{\text{eav}}_{\mathcal{A},\Pi}(n)$ in Figure 60.

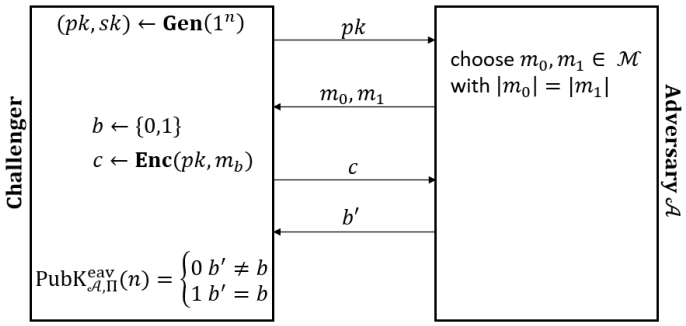


Figure 60: Public-Key IND-EAV Experiment

Definition 4.2 Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$ be a public-key encryption. Define an IND-CPA experiment $\text{PubK}^{\text{cpa}}_{\mathcal{A},\Pi}(n)$ in Figure 61.

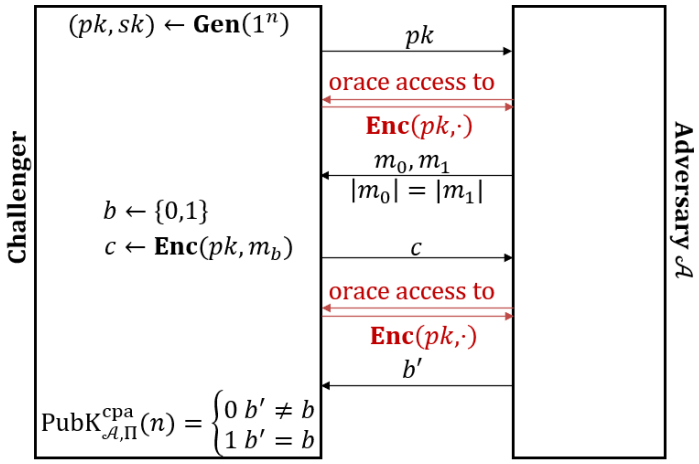


Figure 61: Public-Key IND-CPA Experiment

Theorem 4.1 *For public-key encryption, IND-EAV is equivalent to IND-CPA.*

4.2 EIGamal Encryption

Example 4.1 (Generalized OTP) Let G is an Abelian group of order q, and let $\mathcal{K} = \mathcal{M} = \mathcal{C} = G$. Define a scheme as follows.

- $\mathrm{Gen}(1^n)$: choose $k \leftarrow G$ and output k.
- $\mathrm{Enc}(k, m)$: output $c = k \cdot m$.
- $\mathrm{Dec}(k, c)$: output $k^{-1} \cdot c$.

Theorem 4.2 *The generalized OTP is perfectly secret.*

Proof. For every plaintext m, we have $\Pr[C = c | M = m] = \Pr\left[K = c \cdot m^{-1}\right] = \frac{1}{|G|}$.

Example 4.2 (ElGamal Encryption) Let $\mathcal{G}(1^n)$ be a cyclic group generator. Define a scheme as follows.

- $\mathrm{Gen}(1^n)$:
  - let $(q, G, g) \leftarrow \mathcal{G}(1^n)$;
  - let $x \leftarrow \mathbb{Z}_q$ and $h = g^x$;
  - output $p_k = (q, G, g, h)$ and $sk = x$.
- $\mathrm{Enc}(pk, m)$:
  - let $y \leftarrow \mathbb{Z}_q$ and $c_1 = g^y$;
  - let $c_2 = m \cdot h^y$;
  - output (c_1, c_2).
- $\mathrm{Dec}(sk, c)$: let $c = (c_1, c_2)$ and output c_2 / c_1^x.

Example 4.3 Given subgroup $G = \langle 4 \rangle = \{4, 5, 9, 3, 1\}$ of $\mathbb{Z}_{11}^*$, then we have $p = 11$, $p = 5$ and $g = 4$.

- choose $x = 3$, then $pk = (5, G, 4, 4^3) = (5, G, 4, 9)$ and $sk = 3$.
- given $m = 5$ and choose $y = 2$, then $c = (4^2, 5 \cdot 9^2) = (5, 9)$.

- decrypt $m = 9/5^3$, we have to calculate the inverse of 5^3 such that $5^3 a \equiv 1 \bmod 11$. we can use little Fermat lemma $a \equiv (5^3)^9$.

$$5^3 \equiv 4 \bmod 11$$
$$(5^3)^2 \equiv 5 \bmod 11$$
$$(5^3)^4 \equiv 3 \bmod 11$$
$$(5^3)^8 \equiv 9 \bmod 11$$
$$(5^3)^9 \equiv 3 \bmod 11$$

  Thus $m = 9 \cdot 3 = 5$.

Theorem 4.3 *ElGamal encryption is IND-CPA assume that DDH is hard.*

Proof. Suppose that ElGamal encryption is not IND-CPA. Then there is a PPT adversary $\mathcal{A}$ such that

$$\Pr\left[\text{PubK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right] = \frac{1}{2} + \epsilon$$

for some non-negligible function ϵ (IND-EAV = IND-CPA). We show it is a contradiction by constructing a DDH solver $\mathcal{B}$ that runs $\mathcal{A}$ as subroutine in Figure 62.

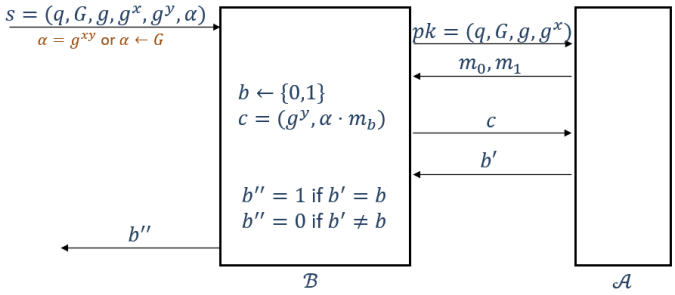


Figure 62: Counterexample DDH solver $\mathcal{B}$

.

Consider two related probability

$$\Pr\left[\mathcal{B}(s) = 1 | \alpha = g^{xy}\right] = \Pr\left[\text{PubK}^{\text{eav}}_{\mathcal{A},\Pi}(n) = 1\right]$$
$$\Pr\left[\mathcal{B}(s) = 1 | \alpha \leftarrow G\right] = \frac{1}{2}$$

Thus the different is ϵ.

4.3 RSA

Example 4.4 (Plain RSA) Let $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) + \mathcal{M}$, where $\mathcal{M} = \{m : m \in [N], gcd(m, N) = 1\}$ and three operations are defined as follows.

- $\text{Gen}(1^n)$:
  - choose two n-bit primes $p \neq q$, let $N = pq$. Then $\phi(N) = (p-1)(q-1)$.
  - choose $e \leftarrow \mathbb{Z}^*_{\phi(N)}$, let $d = e^{-1}$ in $\mathbb{Z}^*_{\phi(N)}$.
  - output $pk = (N, e)$ and $sk = (N, d)$.
- $\text{Enc}(pk, m)$: output $c = m^e \bmod N$.
- $\text{Dec}(sk, c)$: output $m = c^d \bmod N$

The correctness is guaranteed by

$$m^{ed} = m^{1+t\phi(N)} = m$$

We can use extended euclidean algorithm to calculate the inverse of e, by

$$ed = 1 + t\phi(n) \Rightarrow ed - t\phi(n) = 1$$

4.4 Message Authentication Code

The previous MAC scheme has two disadvantages:

- no non-repudiation: the sender of (m, t) can deny the existence of (m, t).
- key distribution and key storage.

Definition 4.3 A digital signature scheme is a tuple $(\text{Gen}, \text{Sign}, \text{Vrfy})$ of three PPT algorithms:

- $(pk, sk) \leftarrow \text{Gen}(1^n)$: pk, for verification; sk for signing.
- $\sigma \leftarrow \text{Sign}(sk, m)$: σ is a signature of m by sk.
- $\{0, 1\} \leftarrow \text{Vrfy}(pk, (m, \sigma))$: output 1 if σ is valid.

Definition 4.4 The signature experiment Sig-forge$_{\mathcal{A},\Pi}(n)$ is defined in Figure 63.

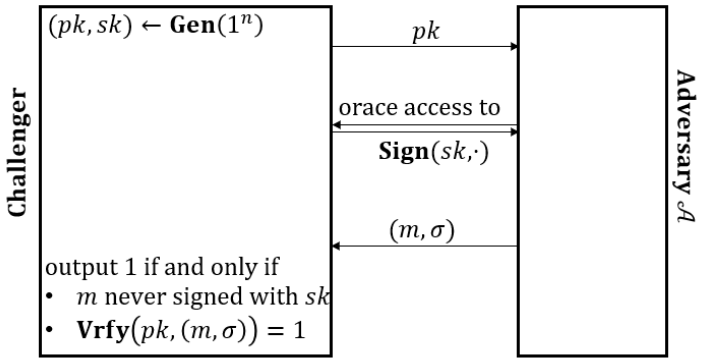


Figure 63: Sig-forge$_{\mathcal{A},\Pi}(n)$ Experiment

.

Definition 4.5 Π is existentially unforgeable under an adoptive chosen message attack (EUF-CMA) if for all PPT adversary $\mathcal{A}$, there is a negligible function negl$(\cdot)$ such that

$$\Pr\left[\text{Sig-forge}_{\mathcal{A},\Pi}(n) = 1\right] \leq \text{negl}(n),$$

where the probability is taken over the random coins of $\mathcal{A}$ and the random coins used in the experiment.