Lecture note 3: Public key cryptography

April 2, 2024

In this lecture, we embark on an exploration of public key cryptography algorithms, starting with a review of Symmetric Key Cryptography. We will then delve into the mathematical foundations that underpin public key cryptography and proceed to the details of its practical implementation. The discussion encompasses prominent methods such as the Diffie-Hellman Key Exchange [DH22], ElGamal encryption [ElG85], and the Rivest-Shamir-Adleman (RSA) algorithm [BB79]. We delve into not only the mechanisms behind these algorithms but also critically assess their security aspects. Finally, we round off our discourse with a comprehensive introduction to secure shell protocol and digital signatures, two other vital applications, that stems from the principles of public key cryptography.

1 Recall Symmetric Key Cryptography

1.1 Symmetric-key Encryption

A symmetric key encryption scheme is proposed in 2002 [DK02]. It consists of a triple of polynomial-time algorithms (Gen, Enc, Dec) and a message space $\mathcal{M}$, key space $\mathcal{K}$, and ciphertext space $\mathcal{C}$.

1. $Gen(1^k)$ is a (randomized) key-generation algorithm that, on input a security parameter k (often represented in unary notation as 1^k), generates key $K \leftarrow \mathcal{K}$, and outputs key pair (K, K).

2. $Enc(K, M)$ is a (randomized) encryption algorithm that, on input a key K and message $M \in \mathcal{M}$, outputs ciphertext $C \in \mathcal{C}$.

3. $Dec(K, C)$ is a deterministic decryption algorithm that, on input a key K and ciphertext $C \in \mathcal{C}$, outputs $M \in \mathcal{M}$ or $\perp$.

We provide a simplified model of symmetric encryption in Figure 3. Note that every part of the encryption process, except for the key itself, is known to the public. In other words, the security of the encryption scheme should not depend on keeping the encryption algorithm secret. This concept aligns with a fundamental principle in cryptography known as Kerckhoffs' principle [Kah96], which states as follows:

> 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.

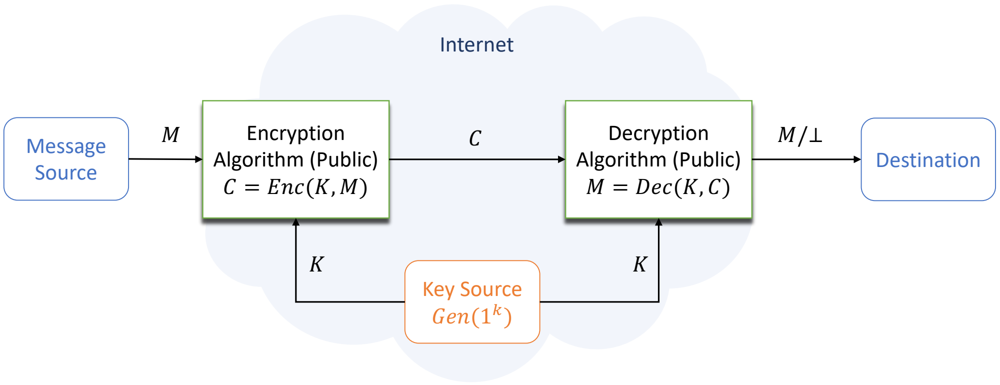

Internet
Message Source
M
Encryption Algorithm (Public) $C = Enc(K, M)$
C
Decryption Algorithm (Public) $M = Dec(K, C)$
$M/\perp$
Destination
K
Key Source $Gen(1^k)$
K

Figure 1: Model of Symmetric-key Encryption

1.2 Security Definitions

Breaking/security is measured by the Aim and Capability of the adversary.

- Aim: Try to learn something meaningful from the target ciphertext.
- Capability:
  - Ciphertext-only attack: the adversary only observes the ciphertext.
  - Chosen-plaintext attack: in addition to observing the ciphertext, the adversary can encrypt messages of his choice.
  - Chosen-ciphertext attack: in addition to observing the ciphertext, the adversary can choose ciphertexts to be decrypted, except for the target ciphertext.

Definition 1. A scheme Π is said to be computationally secure if any probabilistic polynomial time (PPT) adversary succeeds in breaking the scheme with negligible probability.

Here, we give more details about this by introducing indistinguishably-eavesdropper (IND-EAV) security. We define the experiment $\text{Exp}_{\Pi}^{\text{IND-eva}}(\mathcal{A})$ of scheme $\Pi = (Gen, Enc, Dec)$ between the adversary $\mathcal{A}$ and challenger as following,

1. The challenger runs $Gen(1^k)$ to generate the key K.
2. The adversary A outputs a pair of messages $(m_0, m_1) \in \mathcal{M}$ such that $|m_0| = |m_1|$.
3. The challenger flips a fair binary coin $b \leftarrow \{0, 1\}$ and encrypts the message m_b using the key K to obtain the ciphertext $C \leftarrow Enc(K, m_b)$.
4. The adversary A is given the ciphertext C and outputs a guess b'.
5. Return 1 if $b' = b$, and 0 otherwise.

Definition 2 (IND-EAV Security)**.** The IND-EAV-advantage of an adversary $\mathcal{A}$ against IND-EAV security of Π is defined as

$$\text{Adv}_{\Pi}^{\text{IND-EAV}}(\mathcal{A}) = \left| \Pr[\text{Exp}_{\Pi}^{\text{IND-EAV}}(\mathcal{A}) = 1] - \frac{1}{2} \right|$$

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

1.3 Security Proof: Reduction

In computational complexity, the problem A is reducible to the problem B means that solving problem B can be transformed into solving problem A, and this reduction method has been formalized in [GSM18]. This transformation converts a problem instance x_A of problem A into a problem instance x_B of problem B, and converts the problem solution y_B of problem B into the problem solution y_A of problem A. Consequently, this reducibility implies that problem B is at least as difficult as problem A.

1.4 Drawback of Symmetric Key Encryption

The primary disadvantage of using symmetric key encryption in a network of users is the extensive requirement for key management. In a scenario where one user needs to communicate securely with N other users, each pair of users must share a unique symmetric key. This means that for N users, there must be $N(N-1)/2$, which simplifies to $O(N^2)$, different symmetric keys across the network. As the number of participants in the network grows, the number of required keys increases quadratically, leading to significant challenges in securely storing and managing these keys.

2 Introduction to Asymmetric Encryption

Asymmetric cryptography, often referred to as public-key cryptography, is a revolutionary cryptographic paradigm that has significantly shaped the landscape of secure data transmission since its inception. Unlike traditional symmetric cryptography where a single secret key is used for both encryption and decryption, asymmetric cryptography employs a pair of mathematically linked keys: a public key and a private key.

The fundamental principle underlying this system is that information encrypted with one key can only be decrypted using its corresponding unique key. In practice, anyone can use a person's public key to encrypt messages intended for that individual, but the message can only be deciphered by the recipient who possesses the matching private key. This elegant design enables secure communication without requiring a pre-shared secret.

This two-key system provides an enhanced level of security as the public key can be openly shared without compromising the confidentiality of the information being exchanged. Asymmetric cryptography underpins numerous modern security protocols, including digital signatures, key exchange, and secure online transactions such as those facilitated by SSL/TLS certificates.

Two prime examples of asymmetric cryptography algorithms are the ElGamal encryption scheme and the Rivest-Shamir-Adleman (RSA) algorithm, both of which utilize complex mathematical problems to ensure the security of the encryption process. These algorithms have become cornerstones in ensuring the confidentiality, integrity, and authenticity of digital communications across the globe.

3 Related Mathematical Theory

3.1 Modulo

Modulo plays a crucial role in many areas of mathematics, computer science, and cryptography, especially in dealing with finite groups, modular arithmetic, and cyclic structures.

Definition 3. $\forall p \in \mathbb{Z}, q \in \mathbb{Z}\backslash\{0\}$, $\exists k, r \in \mathbb{Z} : p = kq + r, 0 \leq r < q$. In this context, the modulo operation is denoted by

$$p \mod q = r. \tag{1}$$

We usually refer k as $\lfloor \frac{p}{q} \rfloor$.

Definition 4. $\forall q \in \mathbb{Z}\backslash\{0\}, p_1, p_2, ..., p_n \in \mathbb{Z}$, we say $p_1, p_2, ..., p_n$ congruent modulo q iff

$$p_1 \mod q = p_2 \mod q = ... = p_n \mod q \tag{2}$$

, written as:

$$p_1 \equiv p_2 \equiv ... \equiv p_n \pmod q. \tag{3}$$

It's easy to see that $\forall p \in \mathbb{Z}$ we have $p \equiv (p \mod q)(\mod q)$

Lemma 1. If $\forall q \in \mathbb{Z}\backslash\{0\}, p_1, p_2, ..., p_n \in \mathbb{Z}$ and $r_1, r_2, ..., r_n \in \mathbb{Z}$ satisfying $p_i \equiv r_i \mod q$ for all $i \in [n]$, we have

$$\prod_{i=1}^{n} p_i \equiv \prod_{i=1}^{n} r_i \pmod q. \tag{4}$$

Proof. By the definition of congruence, we have:

$$\prod_{i=1}^{n} p_i \equiv \prod_{i=1}^{n} (p_i \mod q) \equiv \prod_{i=1}^{n} (r_i \mod q) \equiv \prod_{i=1}^{n} r_i \pmod q \tag{5}$$

□

3.2 Greatest Common Divisor

Definition 5. Given $p, q \in \mathbb{Z} : (p \neq 0) \vee (q \neq 0)$,

$$\gcd(p, q) := \max\{t : t \in \mathbb{Z}, t|p, t|q\}, \tag{6}$$

which represents the greatest common divisor between p and q.

It's easy to see that $\gcd(p, 0) = p$.

Theorem 1. $\forall p \in \mathbb{Z}, q \in \mathbb{Z}\backslash\{0\}$,

$$\gcd(p, q) = \gcd(p \mod q, q). \tag{7}$$

This is commonly known as Euclidean algorithm.

Proof. By the analysis below:

$$\begin{aligned}
& t|p \wedge t|q \\
\Leftrightarrow & \exists a, b \in \mathbb{Z} \text{ s.t. } at = p \wedge bt = q \\
\Leftrightarrow & \exists a, b \in \mathbb{Z} \text{ s.t. } (a - \lfloor \frac{p}{q} \rfloor b)t = p \mod q \wedge bt = q \\
\Leftrightarrow & \exists a', b \in \mathbb{Z} \text{ s.t. } a't = p \mod q \wedge bt = q \\
\Leftrightarrow & t|(p \mod q) \wedge t|q
\end{aligned}$$

we can see that the the set of common divisors of (p, q) and $(p \mod q, q)$ are the same. Hence, $\gcd(p, q) = \gcd(p \mod q, q)$ by the definition of gcd. □

The Euclidean algorithm proposed in 1844 runs like this [Sha94]:

$$f(p, q) \Rightarrow \begin{cases} \text{output } p, & q = 0 \\ f(q, p \mod q), & \text{otherwise} \end{cases}$$

The correctness comes of Euclidean algorithm from $\gcd(p, 0) = p$ and Theorem 1.

3.3 Bézout's Lemma

Bézout's Lemma is proposed in 1779 [Béz79].

Lemma 2. Suppose $a, b \in \mathbb{Z}$ with $(a \neq 0) \vee (b \neq 0)$. $A := \{xa + yb : x, y \in \mathbb{Z}\}$, $\exists! q \in A$ such that

$$\forall p \in A, q|p. \tag{8}$$

Proof.

Define A as $A(a, b) := \{xa + yb : x, y \in \mathbb{Z}\}$. From the equations below

$$x(a \mod b) + yb = x(a - \lfloor \frac{a}{b} \rfloor \cdot b) + yb = xa + (y - x\lfloor \frac{a}{b} \rfloor)b \overset{y' = y - x\lfloor \frac{a}{b} \rfloor}{=} xa + y'b,$$

we can conclude that $A(a \mod b, b) = A(a, b)$. Then, we apply the same technique from Euclidean algorithm, obtaining the fact that $A(a, b) = A(\gcd(a, b), 0)$. Now for $A = A(\gcd(a, b), 0)$, it's easy to see that $\gcd(a, b)$ is the only q satisfying $\forall p \in A, q|p$.

□

Lemma 3. Given $a, b \in \mathbb{Z}$ with $(a \neq 0) \vee (b \neq 0)$, $\exists x, y \in \mathbb{Z}$ such that

$$\gcd(a, b) = xa + yb. \tag{9}$$

Proof. By Lemma 2, $\gcd(a, b) \in A(a, b) = \{xa + yb : x, y \in \mathbb{Z}\}$. □

3.4 Group Theory

The group theory can be found in [Cay54].

Definition 6. A set G with a binary operation "$\cdot$" is an algebraic structure, if it is closure, i.e., $\forall a, b \in G$, $a \cdot b \in G$.

By default, we consider this binary operation as multiplication and omit the multiplication symbol "$\cdot$" during computation.

Definition 7. An algebraic structure G is a semigroup, if it has associative property, i.e., $\forall a, b, c \in G$, $(ab)c = a(bc)$.

Definition 8. A semigroup G is a monoid, if it has an identity element, denoted as e, such that $\forall a \in G$, $ae = ea = a$.

When the binary operation is "$+$", we commonly refer to this element as the zero element.

Definition 9. A monoid G is a group, if any elements of G has an inverse element, i.e., $\forall a \in G$, $\exists b \in G$, such that $ab = ba = e$.

Lemma 4. Uniqueness of an Inverse: $\forall a \in G$, $\exists! b \in G$, such that $ab = ba = e$, if G is a group.

Proof. Giving a group G, $\forall a \in G$, $\exists b, c \in G$ such that $ab = ba = e, ac = ca = e$, and then

$$b = be = b(ac) = (ba)c = ec = c. \tag{10}$$

□

Lemma 5. Uniqueness of Operation: $\forall a \in G$, $\exists! b \in G$, such that $ab = c$, if G is a group.

Proof. Giving a group G, $\forall a \in G$, $\exists b, c, d \in G$ such that $ab = ba = e, ac = ad$, and then

$$c = ec = bac = bad = ed = d. \tag{11}$$

□

Definition 10. Given a group G, the order of G, denoted by $|G|$, is the number of elements in the group. This is either a finite number or is infinite.

3.5 Exponential Notation

Definition 11. $\forall g \in G$, $g^0 = e$.

Definition 12. $\forall g \in G$, $gg^{-1} = g^{-1}g = e$.

Definition 13. $\forall g \in G$, $\forall n \in \mathbb{Z}$, $\exists r \in \{-1, 1\}$ such that $n = r|n|$, $g^n = g^{r|n|} = \underbrace{g^r....g^r}_{|n|}$.

Lemma 6. $\forall g \in G$, $\forall n, m \in \mathbb{Z}$, $g^n g^m = g^{n+m}$.

Proof.
1. $\forall n \in \mathbb{Z}, m = 0 \Rightarrow g^n g^0 = g^n e = g^n$, and vice versa.
2. $\forall n, m \in \mathbb{Z}$, $\exists r, s \in \{-1, 1\}$ such that $n = r|n|, m = s|m|$ and then

$$g^n g^m = g^{r|n|} g^{s|m|} = \underbrace{g^r....g^r}_{|n|}\underbrace{g^s....g^s}_{|m|} = g^{r|n|+s|m|} == g^{n+m}. \tag{12}$$

□

Lemma 7. $\forall g \in G$, $\forall n, m \in \mathbb{Z}$, $g^{nm} = (g^n)^m$.

Proof.
1. $g^{-1}(g^{-1})^{-1} = e = g^{-1}g \Rightarrow (g^{-1})^{-1} = g \Rightarrow (g^{-1})^{-1} = g^{(-1)\cdot(-1)}$
2. $\forall n, m \in \mathbb{Z}$, $\exists r, s \in \{-1, 1\}$ such that $n = r|n|, m = s|m|$ and then

$$\begin{aligned} g^{nm} = g^{rs|n||m|} &= \underbrace{g^{rs}....g^{rs}}_{|n||m|} = \underbrace{\underbrace{(g^r....g^r)^s}_{|n|}}_{|m|} \\ &= (g^{r|n|})^{s|m|} = (g^n)^m \end{aligned} \tag{13}$$

□

3.6 Subgroup

Definition 14. Let H be a subset of group G, we say that H is a subgroup iff H forms a group with the operation in G, denoted as $H \leq G$.

Definition 15. Let $H \leq G$, if $g \in G$:

- The right coset of H generated by g is $Hg = \{hg : h \in H\}$.
- The left coset of H generated by g is $gH = \{gh : h \in H\}$.

Definition 16. A set $H_1H_2 := \{ab : a \in H_1, b \in H_2\}$ is the product of subgroup if $\forall H_1, H_2 \leq G$, which does not need to be a subgroup of G.

Definition 17. Let $H \leq G$. H is a normal subgroup of G if any of the following holds:

- $cHc^{-1} \subseteq H$ for all $c \in G$.
- $cHc^{-1} = H$ for all $c \in G$.
- $cH = Hc$ for all $c \in G$.
- Every left coset of H in G is also a right coset.
- Every right coset of H in G is also a left coset.

Written as $H \trianglelefteq G$.

Definition 18. if $H \trianglelefteq G$, the quotient group $\frac{G}{H}$ is defined as $\frac{G}{H} := \{aH : a \in G\}$.

Lemma 8. $\forall H \leq G, \forall a \in G$, aH or Ha is a group if and only if $aH = Ha = H$.

Proof.
1. *Sufficiency.* $\forall H \leq G \Rightarrow e \in H \Rightarrow \forall a \in G, a = ae \in aH$. Since aH is a group, $e \in aH \Rightarrow aa^{-1} \in aH \Rightarrow a^{-1} \in H$. Since H is a group, $a^{-1} \in H \Rightarrow a \in H \Rightarrow \forall b \in H, ab, ba \in H \Rightarrow aH = Ha = H$.
2. *Necessity.* $\forall H \leq G, \forall a \in G, a \in H$. Since H is a group, $aH = Ha = H$ is a group. □

Lemma 9. $\forall H \leq G, \forall a \in G$, $|aH| = |H|$.

Proof. $\forall H \leq G$, $\forall a \in G, \exists b, c \in H$ such that $b \neq c$,

$$b, c \in G \Rightarrow ab \neq ac, \tag{14}$$

due to the uniqueness of group operation (Lemma 5). Thus, $|aH| = |H|$. □

Lemma 10. $\forall H \leq G, G = \bigcup\limits_{a \in G} aH = \bigcup\limits_{a \in G} Ha$.

Proof. $\forall H \leq G \Rightarrow e \in H \Rightarrow \forall a \in G, a \in aH \Rightarrow \{a\} \subseteq aH \Rightarrow G = \bigcup\limits_{t} \{t\} \subseteq \bigcup\limits_{t \in G} tH$. Due to closure of G, $\forall b \in aH \Rightarrow b \in G \Rightarrow \bigcup\limits_{t \in G} tH \subseteq G$. Finally, $G = \bigcup\limits_{t \in G} tH$ and similarly $G = \bigcup\limits_{t \in G} Ht$ □

Lemma 11. $\forall H \leq G, \forall a, b \in G$, such that

$$aH \cap bH \neq \emptyset \Leftrightarrow aH = bH. \tag{15}$$

Proof.
1. *Necessity.* $aH = bH \Rightarrow aH \cap bH = aH \cap aH = aH \neq \emptyset$.
2. *Sufficiency.* $aH \cap bH \neq \emptyset \Rightarrow \exists x \in aH \cap bH$. $\exists h_1, h_2 \in H$, such that $x = ah_1 = bh_2$, which implies that

$$b = ah_1h_2^{-1}, a^{-1}b = h_1h_2^{-1}. \tag{16}$$

Since H is a group, $\forall h \in H \Rightarrow hH = H$. Thus, $bH = ah_1h_2^{-1}H = a(h_1(h_2^{-1}H)) = aH$ □

Theorem 2. Lagrange's theorem: $\forall H \leq G$,

$$|G| = [G, H]|H|. \tag{17}$$

where $[G, H]$ is the number of left cosets of H in G [Lag71].

Proof. By Lemma 11. $\forall F_1, F_2 \in \frac{G}{H}$, such that $F_1 \neq F_2$,

$$F_1 \cap F_2 = \emptyset \Rightarrow |F_1 \cup F_2| = |F_1| + |F_2|. \tag{18}$$

Then, we have

$$|G| = \left| \bigcup_{F_i \in \frac{G}{H}} F_i \right| = \sum_{F_i \in \frac{G}{H}} |F_i|, \tag{19}$$

where $|F_i| = |H|$ due to the uniqueness of group operation (Lemma 5), so that

$$|G| = \sum_{F_i \in \frac{G}{H}} |H| = |\frac{G}{H}||H| = [G, H]|H| \tag{20}$$

□

3.7 Generating Group

Definition 19. $\forall g \in G$, $\{g^i | i \in \mathbb{Z}\}$ is the generating set of G, denoted as $\langle g \rangle$, where g is the generator of the generated set.

Lemma 12. *$\forall g \in G$, $\langle g \rangle \subseteq G$.*

Proof. $\forall a \in \langle g \rangle$, $\exists n \in \mathbb{Z}$ such that

$$a = g^n. \tag{21}$$

Since G is closed, $g^n \in G \Rightarrow a \in G$. □

Lemma 13. *$\forall g \in G$, $\forall a, b \in \langle g \rangle$, $ab \in \langle g \rangle$.*

Proof. $\forall a, b \in \langle g \rangle$, $\exists n, m \in \mathbb{Z}$ such that $a = g^n, b = g^m$, and then

$$n + m \in \mathbb{Z} \Rightarrow ab = g^n g^m = g^{n+m} \in \langle g \rangle . \tag{22}$$

□

Lemma 14. *$\forall g \in G$, $\forall a, b, c \in \langle g \rangle$, $abc = a(bc)$.*

Proof. $\forall a, b, c \in \langle g \rangle$, $\exists n, m, l \in \mathbb{Z}$ such that $a = g^n, b = g^m, c = g^l$, and then

$$abc = g^n g^m g^l = g^{n+m+l} = g^{n+(m+l)} = g^n(g^m g^l) = a(bc). \tag{23}$$

□

Lemma 15. *$\forall g \in G$, $\exists e \in \langle g \rangle$ such that $\forall a \in \langle g \rangle , ea = ae = a$.*

Proof. $\forall n \in \mathbb{Z}$, we have $g^0, g^n \in \langle g \rangle$ and then, by Lemma 6,

$$g^0, g^n \in G \Rightarrow g^0 g^n = g^n, g^n g^0 = g^n. \tag{24}$$

□

Lemma 16. *$\forall g \in G$, $\forall a \in \langle g \rangle$, $\exists b \in \langle g \rangle$ such that $ab = e$.*

Proof. $\forall n \in \mathbb{Z}$, $\exists m \in \mathbb{Z}$ such that $n + m = 0$ and then, by Lemma 6,

$$g^n g^m = g^{n+m} = g^0 = e. \tag{25}$$

□

Theorem 3. *$\forall g \in G$, $\langle g \rangle$ is a group.*

Proof. According to Lemma 13, Lemma 14 Lemma 15, and Lemma 16, $\langle g \rangle$ satisfies Definition 6, Definition 7, Definition 8, Definition 9, respectively. □

Theorem 4. *$\forall g \in G$, $\langle g \rangle \leq G$.*

Proof. According to Theorem 3 and Lemma 12, $\langle g \rangle$ satisfies Definition 14. □

3.8 Finite Group

Definition 20. A group G is a finite group, if $|G| \in \mathbb{Z}^+$.

Lemma 17. $\forall g \in G$, $\langle g \rangle$ is a finite group, if G is a finite group.

Proof. $\forall g \in G, \langle g \rangle \leq G \Rightarrow \langle g \rangle \subseteq G \Rightarrow |\langle g \rangle| \leq |G|$ and then, $G \in \mathbb{Z}^+ \Rightarrow |\langle g \rangle| \in \mathbb{Z}^+$ □

Lemma 18. $\forall g \in G$, $\exists n \in \mathbb{Z}^+$ such that $g^n = e$, if G is a finite group.

Proof. Since G is closed and finite, $\exists i, n \in \mathbb{Z}^+$ such that g^i repeats in a sequence of exponential g:

$$< g^0, g^1, ..., g^i, \qquad ..., g^{i-1}, \quad g^i, ... > \tag{26}$$

$$= < g^0, g^1, ..., g^{i+(1-1)}, ..., g^{i+(n-1)}, g^i, ... >, \tag{27}$$

where $g^{i-1} = g^{i+(n-1)} = g^{i-1+n} = g^{i-1}g^n \Rightarrow g^n = e$. □

Theorem 5. Cyclic group: $\forall g \in G$, $\langle g \rangle$ is cyclic, i.e., $g^{|\langle g \rangle|} = e$, if $\langle g \rangle$ is a finite group.

Proof. $\forall g \in G, n := \min\{t : t \in \mathbb{Z}^+, g^t = e\}$. In this context, $\forall m \in \mathbb{Z}, \exists k, r \in \mathbb{Z}, 0 \leq r < n$, such that $m = kn + r$, and then

$$g^m = g^{kn+r} = g^{kn}g^r = eg^r = g^r. \tag{28}$$

Here, $0 \leq r < n$, which implies that there are n elements in $\langle g \rangle$, i.e, $|\langle g \rangle| = n \Rightarrow g^{|\langle g \rangle|} = g^n = e$. □

Theorem 6. $\forall g \in G, g^{|G|} = e$, if G is a finite group.

Proof. $\forall g \in G$:

1. $\langle g \rangle$ is a finite group and $g^{|\langle g \rangle|} = e$ by Lemma 17 and Theorem 5;
2. $\langle g \rangle \leq G$ by Theorem 4.

By Lagrange's Theorem (Theorem 2), therefore, $\exists k \in \mathbb{Z}^+$ such that

$$|G| = k|\langle g \rangle|, \tag{29}$$

where $k = [G, \langle g \rangle]$ is the number of coset of $\langle g \rangle$ on G. Sequentially, we have $g^{|G|} = g^{k|\langle g \rangle|} = (g^{|\langle g \rangle|})^k = e^k = e$ and proves our claim. □

Corollary 1. $\forall H \leq G$ are finite groups, if G is a finite group.

Proof. By Lagrange's Theorem (Theorem 2), therefore, $\exists k \in \mathbb{Z}^+$ such that

$$|G| = k|H|, \tag{30}$$

where $k = [G, H]$ is the number of coset of H on G. Thus, $|H|$ is not infinite when $|G| \in \mathbb{Z}^+$ and H is a finite group. □

Theorem 7. $\forall H \leq G$, $\forall a \in H$, $a^{|G|} = e$, if G is a finite group.

Proof. By Theorem 6, $\forall a \in H$,

$$a \in G \Rightarrow a^{|G|} = e. \tag{31}$$

□

Corollary 2. Given a finite group G, $\forall n \in \mathbb{Z}$, $g^n = g^{n \mod |G|}$.

Proof. $\forall n \in \mathbb{Z}, \exists k, r \in \mathbb{Z}, 0 \leq r < |G|$, such that

$$n = k|G| + r \Rightarrow g^n = g^{k|G|+r} = g^{k|G|}g^r = eg^r = g^r, \tag{32}$$

where $r = n \mod |G| \Rightarrow g^n = g^{n \mod |G|}$. □

3.9 Abelian Group

Definition 21. A group G is an Abelian group, if it satisfies commutativity: $\forall a, b \in G$, the equation $ab = ba$ holds.

Corollary 3. $\forall H \leq G$ are Abelian groups, if G is an Abelian group.

Proof. $\forall a, b \in H$, since H is a subgroup and therefore closed under the group operation, it follows that:

$$ab, ba \in H. \tag{33}$$

Moreover, since H is a subset of the Abelian group G, we have

$$a, b \in G \Rightarrow ab = ba. \tag{34}$$

This verifies that H is an Abelian group. □

Corollary 4. $\forall H_1, H_2 \leq G$, H_1H_2 is an Abelian group, if G is an Abelian group.

Proof. $\forall a, b \in H_1H_2$, $\exists a_1, b_1 \in H_1, a_2, b_2 \in H_2$ such that $a = a_1a_2, b = b_1b_2$ and then, by commutativity of G, we have

$$ab = a_1a_2b_1b_2 = (a_1b_1)(a_2b_2) \in H_1H_2, \tag{35}$$

which implies the property of closure holds. Moreover, H_1H_2 inherits the associative property, the commutativite property, and the identity element from G. Next, let's prove the invertibility of all elements in H_1H_2.

$\forall a \in H_1H_2$, $\exists a_1, b_1 \in H_1, a_2, b_2 \in H_2$ such that $a = a_1a_2$ and $a_1b_1 = e, a_2b_2 = e$ and then, by commutativity of G, we have

$$ab = a_1a_2b_1b_2 = (a_1b_1)(a_2b_2) = ee = e \in H_1H_2, \tag{36}$$

and thus we prove our claim. □

3.10 Modulo Multiplication Group

Definition 22. $\forall x, y, n \in \mathbb{Z}^+$,

$$\mathbb{Z}_n^* := (\{t : t \in \mathbb{Z}^+, t < n, \gcd(t, n) = 1\}, *), \tag{37}$$

where $x * y := x \cdot y \mod n$.

Lemma 19. $\mathbb{Z}_n^*$ is a closure algebraic structure.

Proof. $\forall a, b \in \mathbb{Z}_n^*$,

$$c := a * b = ab \mod n. \tag{38}$$

Obviously, $c \in \mathbb{Z}$ and $c < n$. To show closure, we must demonstrate that $\gcd(c, n) = 1$. By Bézout's Lemma (see in Sec. 3.3), $\exists x_1, x_2, y_1, y_2 \in \mathbb{Z}$, such that

$$x_1 a + y_1 n = 1, x_2 b + y_2 n = 1. \tag{39}$$

Multiplying these two equations together yields:

$$(x_1 x_2)ab + (x_1 a + x_2 b + y_1 y_2 n)n = 1, \tag{40}$$

which implies $\gcd(ab, n) = 1$ by virtue of Bézout's Lemma. Applying the properties of the Euclidean algorithm (Theorem 1):

$$\gcd(c, n) = \gcd(ab \mod n, n) = \gcd(ab, n) = 1, \tag{41}$$

we prove our claims. □

Lemma 20. $\mathbb{Z}_n^*$ is associative.

Proof. $\forall a, b, c \in \mathbb{Z}$,

$$\begin{aligned}(a * b) * c =&(ab \mod n)c \mod n \\ =&abc \mod n \\ =&a(bc \mod n) \mod n = a * (b * c)\end{aligned} \tag{42}$$

□

Definition 23. The identity element of $\mathbb{Z}_n^*$ is $1 \mod n$.

Lemma 21. Every element of $\mathbb{Z}_n^*$ has an inverse element: $\forall a \in \mathbb{Z}_n^*, \exists a^{-1} \in \mathbb{Z}_n^*$, such that

$$a * a^{-1} = 1. \tag{43}$$

Proof. $\forall n \in \mathbb{Z}$, $\forall a \in \mathbb{Z}_n^*$, we have

$$\gcd(a, n) = 1. \tag{44}$$

According to 3.3, therefore, $\exists b, k \in \mathbb{Z}$, such that

$$ab + kn = \gcd(a, n) = 1, \tag{45}$$

which implies

$$ab \mod n = 1 \Rightarrow a * b = 1, \tag{46}$$

where b is the inverse of a, i.e. $a^{-1} = b$. □

Lemma 22. $\mathbb{Z}_n^*$ is commutativite.

Proof. $\forall x, y \in \mathbb{Z}, xy \mod n = yx \mod n \Rightarrow x * y = y * x$. □

Theorem 8. $\mathbb{Z}_n^*$ is an Abelian group.

Proof. According to Lemma 19, Lemma 20, Definition 23, Lemma 21, and Lemma 22, $\mathbb{Z}_n^*$ satisfies Definition 6, Definition 7, Definition 8, Definition 9 and Definition 21, respectively. □

Theorem 9. $\mathbb{Z}_n^*$ is a finite group.

Proof. $|\mathbb{Z}_n^*| < n$, which satisfies Definition 20. □

Corollary 5. $\forall a \in \mathbb{Z}_n^*$, $\forall i \in \mathbb{Z}$, $a^i = a^{i \mod |\mathbb{Z}_n^*|}$.

Proof. The claim is proved by Theorem 9 and Corollary 2. □

Definition 24. Euler's totient function, denoted by $\varphi(n)$, is the number of positive integers less than n that are coprime to n, i.e., $\varphi(n) := |\mathbb{Z}_n^*|$.

Corollary 6. $\exists p_1, ..., p_n \in \mathbb{Z}^+$ such that $\gcd(p_i, p_j) = 1$ where $i, j = 1, ..., n$ and $i \neq j$ and then

$$\varphi(p_1...p_n) = \prod_{i=1}^{n} \varphi(p_i). \tag{47}$$

Proof. Giving $i, j \in \mathbb{Z}^+$ with $i \neq j$, let $a \in \mathbb{Z}$ with $1 \leq a \leq p_i p_j$ and $\gcd(a, p_i p_j) = 1$, $\exists x, y \in \mathbb{Z}$ such that

$$xa + y p_i p_j = 1 \Rightarrow \gcd(a, p_i) = 1 \wedge \gcd(a, p_j) = 1. \tag{48}$$

Let $\bar{P}_t := \{p : p \in \mathbb{Z}, 1 \leq p \leq p_i p_j, \gcd(p, p_t) \neq 1\}$ where $t = i, j$, then we have

$$\bar{P}_j = \{p_i p_j\} \Rightarrow |\bar{P}_i \cup \bar{P}_j| = |\bar{P}_i| + |\bar{P}_j| - |\bar{P}_i \cap \bar{P}_j| = p_i + p_j - 1, \tag{49}$$

and thus $\varphi(p_i p_j) = p_i p_j - |\bar{P}_i \cup \bar{P}_j| = p_i p_j - p_i - p_j + 1 = (p_i - 1)(p_j - 1) = \varphi(p_i)\varphi(p_j)$, which implies that φ is multiplicative when $\gcd(p_i, p_j) = 1$. Generally, the equation

$$\varphi(p_1...p_n) = \varphi(p_1...p_{n-1})\varphi(p_n) = \varphi(p_1...p_{n-2})\varphi(p_{n-1})\varphi(p_n) = ... = \prod_{i=1}^{n} \varphi(p_i) \tag{50}$$

holds if any two number p_i and p_j are coprime. □

Corollary 7. $\forall p \in \mathbb{P}, \forall k \in \mathbb{Z}+$,

$$\varphi(p^k) = p^{k-1}(p - 1). \tag{51}$$

Proof. let $a \in \mathbb{Z}$ with $1 \leq a \leq p^k$ and $\gcd(a, p_k) = 1$, $\exists x, y \in \mathbb{Z}$ such that

$$xa + y p^{k-1} p = 1 \Rightarrow \gcd(a, p) = 1. \tag{52}$$

So that we have $P := \{p, 2p, ..., p^{k-1}p\}$ as the set of all of a, where $|P| = p^{k-1} \Rightarrow \varphi(p^k) = p^k - p^{k-1} = p^{k-1}(p - 1)$. □

Corollary 8. $\forall p_1, ..., p_n \in \mathbb{P}$,

$$\varphi(p_1^{k_1}, ..., p_n^{k_n}) = \prod_{i=1}^{n} p_i^{k_i - 1}(p_i - 1). \tag{53}$$

Proof. Let $p_i, p_j \in \mathbb{P}$ with $\gcd(p_i, p_j) = 1, i, j = 1, ..., n, i \neq j$, $\exists x, y \in \mathbb{Z}$ such that

$$xp_i + yp_j = 1 \Rightarrow x(p_i)p_i^{k_i} + y(p_j)p_j^{k_j} = 1, \tag{54}$$

where $x(p_i), y(p_j)$ are polynomials of p_i, p_j, respectively, which implies that $\gcd(p_i^{k_i}, p_j^{k_j}) = 1$. By Corollary 6, $\varphi(p_1^{k_1}, ..., p_n^{k_n}) = \prod_{i=1}^{n} \varphi(p_i^{k_i})$ holds and since Corollary 7, it follows that $\varphi(p_1^{k_1}, ..., p_n^{k_n}) = \prod_{i=1}^{n} \varphi(p_i^{k_i}) = \prod_{i=1}^{n} p_i^{k_i - 1}(p_i - 1)$. □

Theorem 10. Euler's theorem: $\forall a, n \in \mathbb{Z}^+$ such that $\gcd(a, n) = 1$,

$$a^{\varphi(n)} = 1 \pmod{n}. \tag{55}$$

Proof. $\forall a, n \in \mathbb{Z}^+$, $\exists k, r \in \mathbb{Z}$ such that $0 \leq r < n$ and then

$$a = kn + r, \tag{56}$$

where $\gcd(r, n) = 1$ due to Bézout's Lemma 3.3, which implies that $r \in \mathbb{Z}_n^*$. By Definition 24 and Corollary 5, we know that

$$r^{\varphi(n)} = r^{|\mathbb{Z}_n^*|} = 1. \tag{57}$$

Applying a lemma regarding multiplication modulo (Lemma 1), it follows that

$$a^{\varphi(n)} = (kn + r)^{\varphi(n)} = r^{\varphi(n)} \pmod{n} = 1 \pmod{n}. \tag{58}$$

□

4 Hard problems/assumptions in PKE

In this subsection we mainly focus on 4 assumptions that are used in PKE: Decisional Diffie–Hellman (DDH for short) assumption, Discrete logarithm(DL for short) assumption, RSA assumption, and Factorization assumption.

DDH assumption [Bon98]: Consider a cyclic group G with prime order q, and a generator g. The DDH assumption states that g^{ab} is indistinguishable from a random element g^c of G. The formal definition is as follows:

Definition 25.

$$(G, q, g, g^a, g^b, g^{ab}) \overset{c}{\approx} (G, g, q, g^a, g^b, g^c)$$

where (G, q, g) is the group description, $a, b, c \overset{\$}{\leftarrow} Z_q$, and $\overset{c}{\approx}$ states for computational indistinguishability in $|q| = \lambda$.

DL assumption [BG04]: Consider a cyclic group G with order q, and a generator g. The DL assumption states that given a random element $g^a \in G$, it's hard to output the exponent a.

Definition 26. $\forall$ PPT A, the probability

$$\Pr[A(G, q, g, g^a) = a | a \overset{\$}{\leftarrow} Z_q]$$

is negiligible in $|q|$, where (G, q, g) is the group description for a random group with prime order $|q| = \lambda$.

Factorization assumption [CP05]: Consider N where $N = pq$, and p, q are λ-bit primes. The factorization assumption states that given N, it's hard to output p, q such that $N = pq$.

Definition 27. $\forall$ PPT A, the probability

$$\Pr[A(N) = (p, q) | p, q \text{ are } \lambda\text{-bit primes}]$$

is negiligible in λ.

RSA assumption [BV98]: Given N, an integer $e > 0$ that is prime to $\phi(N)$, and an element $y \in \mathbb{Z}_N^*$ where $N = pq$, p, q are λ-bit primes, and $\phi(N) = (p-1)(q-1)$, it's hard to find x such that $x^e = y \mod N$.

Definition 28. $\forall$ PPT A, the probability

$$\Pr[A(N, e, y) = x | x \overset{\$}{\leftarrow} Z_N^*]$$

is negiligible in λ.

5 Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange is a cryptographic protocol that enables two parties to establish a shared secret key over an insecure communication channel. This groundbreaking method, invented by Whitfield Diffie and Martin Hellman in 1976, revolutionized the field of cryptography [DH22]. In a Diffie-Hellman key exchange, both parties involved generate public-private key pairs and exchange their public keys. Without revealing their private keys, they can jointly compute a shared secret which can then be used as the basis for encrypting subsequent communications.

5.1 Method

The Diffie-Hellman key exchange process involves several steps to securely establish a shared secret between two parties, let's call them Alice and Bob. Here's a detailed description of the process:

1. Both Alice and Bob agree on public parameters (q, g), where q is a chosen order used to define a modulo multiplication group $\mathbb{Z}_q^*$ and $g \in \mathbb{Z}_q^*$ is the generator for a subgroup, $\langle g \rangle$, of $g \in \mathbb{Z}_q^*$.

2. Alice generates a random private integer $1 < a < |\mathbb{Z}_q^*|$, computes $A = g^a$, and keeps a secret.

3. Alice sends (q, g, a) to Bob.

4. Bob receives (q, g, a) and generates his own random private integer $1 < b < |\mathbb{Z}_q^*|$, computes $B = g^b$, and also keeps b secret. Then, Bob sends B to Alice and computes shared key by $K = A^b = g^{ab}$.

5. Alice receives B and computes the shared key by $K = B^a = g^{ba} = g^{ab}$ (by the commutativity of $\mathbb{Z}_p^*$, see in Lemma 22).

The shared secret can now be used as the key for a symmetric encryption algorithm, allowing both Alice and Bob to communicate securely, even though the channel they use to exchange messages may be eavesdropped upon.

5.2 Correctness and security

Correctness is quite straightforward since both Alice and Bob obtain the same g^{ab} as the symmetric key.

the DL assumption ensures that only Alice has the value of a, and only Bob has the value of b.

Furthermore, for any other users, the DDH assumption ensures that the key g^{ab} appears as a random group element in their view; even if they can obtain results encrypted with this key, they will not learn any information about the message unless Alice or Bob voluntarily discloses it.

6 Public key encryption

Public key encryption(PKE for short), or asymmetric cryptography, is a cryptographic system that uses two different keys for encrypting and decrypting data:

- Public Key: This key is made publicly available to anyone who wants to send a secure message to the key owner. It is used to encrypt data, meaning anyone with the public key can encrypt information, but they cannot decrypt it.

- Private Key: This key remains confidential and is only known by the owner. It is used to decrypt data that has been encrypted with the matching public key.

In symmetric encryption, the same key is used for both encrypting and decrypting the data. This poses a significant challenge in securely distributing the key to the intended recipient without interception by unauthorized parties. PKE eliminates this problem because the public key can be openly distributed, while the private key remains secure with the owner.

A PKE scheme composes mainly 3 algorithms:

- KeyGen(λ): Input the security parameter $\lambda \in \mathbb{N}$, output a secret-public key pair $(sk, pk) \in \mathcal{SK} \times \mathcal{PK}$. The public key is subsequently made public, while the private key is retained only by the creator of the key pair.

- Enc(pk, M): Input the public key $pk \in \mathcal{PK}$ and a plaintext $M \in \mathcal{M}$, output a ciphertext $C \in \mathcal{C}$ according to M and pk.

- Dec(sk, C): Input the secret key $sk\mathcal{SK}$ and a ciphertext $C \in \mathcal{C}$, output M s.t. C is a correct ciphertext generated by Enc(pk, M).

Where $\mathcal{PK}$, $\mathcal{SK}$, $\mathcal{M}$ and $\mathcal{C}$ represents the public key space, the private key space, the plaintext message space and the ciphertext space. In subsequent discussions, we'll omit λ and treat it as a default parameter.

6.1 Defining correctness and security of PKE

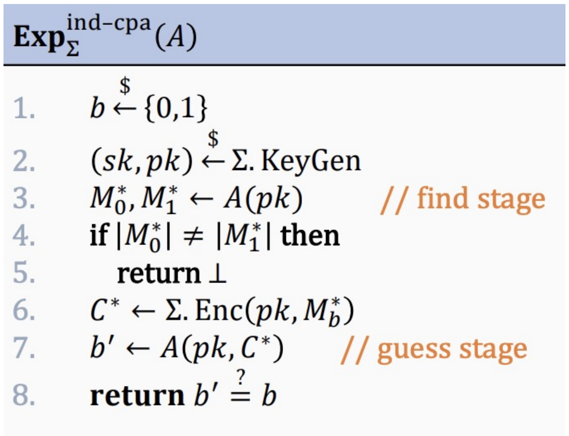
To ensure the correctness of PKE, we want to guarantee that a message, after being encrypted into ciphertext using the public key, will yield the same original message when the corresponding ciphertext is decrypted using the private key. In a formal description:

Definition 29. Correctness:

$$\forall M \in \mathcal{M}, \Pr[\mathrm{Dec}(sk, \mathrm{Enc}(pk, M)) = M] = 1$$

where $(sk, pk) \leftarrow$ KeyGen.

The security of PKE aims to ensure that the ciphertext does not reveal any information about the plaintext message, not even a single bit. We use the following experiment to capture the leakage of 1-bit information: This experiment requires the adversary to determine which of the two chosen plaintext messages corresponds to the encryption, thus representing 1 bit of information. If 1 bit is not leaked, then the adversary only has a 1/2 probability of making the correct determination. Therefore, our formal definition of security is as follows:



```
$\mathbf{Exp}_{\Sigma}^{\text{ind-cpa}}(A)$
$b \overset{\$}{\leftarrow} \{0,1\}$
$(sk, pk) \overset{\$}{\leftarrow} \Sigma.\mathrm{KeyGen}$
$M_0^*, M_1^* \leftarrow A(pk)$        // find stage
if $|M_0^*| \neq |M_1^*|$ then
    return $\perp$
$C^* \leftarrow \Sigma.\mathrm{Enc}(pk, M_b^*)$
$b' \leftarrow A(pk, C^*)$        // guess stage
return $b' \overset{?}{=} b$
```

Figure 2: IND-CPA experiment

Definition 30. The PKE scheme $\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ the is IND-CPA-secure if $\forall$ PPT adversary A, the IND-CPA-advantage (IND-CPA refers to indistinguishability under chosen plaintext attack) of A defined as

$$\mathbf{Adv}_{\Sigma}^{\text{ind-cpa}} = |2 \cdot \Pr[\mathbf{Exp}_{\Sigma}^{\text{ind-cpa}}(A) \Rightarrow 1] - 1|$$

is negligible.

6.2 ElGamal Encryption

ElGamal Encryption, a public-key cryptosystem developed by Taher Elgamal in 1985, is a fundamental cryptographic technique that provides secure data transmission based on the Diffie-Hellman key exchange [ElG85]. It employs a non-secretive key generation process where each participant generates and publicly shares a key pair consisting of a public key and a private key. The algorithm's security relies on the difficulty of the discrete logarithm problem, which makes it computationally infeasible for an attacker to derive the private key from the public one.

In contrast to symmetric encryption methods where the same secret key is used for both encryption and decryption, ElGamal Encryption uses different keys for these processes. A sender can encrypt messages using the recipient's public key, while the recipient can only decrypt them with their corresponding private key. This feature enables secure communication between parties without requiring a pre-shared secret, thus facilitating secure information exchange over insecure channels.

ElGamal Encryption also has the advantage of being malleable, meaning it allows operations like homomorphic addition of ciphertexts. However, its primary use is often as a building block for hybrid encryption systems, combining the efficiency of symmetric ciphers with the key distribution benefits of public-key cryptography.

6.2.1 Method

The ElGamal Encryption is a public-key cryptosystem. It has few public parameters:

1 $\mathbb{Z}_p$: prime field with modular p satisfying $|q| = poly(\lambda)$.

2 $\mathbb{G}_q$: a cyclic subgroup of $\mathbb{Z}_p$ with order q satisfying $|q| = \lambda$.

3 g: the multiplicative generator of $\mathbb{G}_q$.

Now we formally describe the algorithms (KeyGen,Enc,Dec):

- KeyGen: Output the secret key $b \leftarrow \mathbb{Z}_q^*$ and the public key $B = g^b$.
- Enc(B, M): Input the public key B and a message $M \in \mathbb{Z}_q^*$, this algorithm generates $a \leftarrow \mathbb{Z}_q^*$, $A := g^a$ and $K := B^a$, then outputs ciphertext $(A, C := K \cdot M)$.
- Dec$(b, (A, C))$: Input the secret key b and ciphertext C, this algorithm generates $K := A^b$ and outputs $M' := C/K$.

6.2.2 Correctness and security

Correctness:

$$C/K = (B^a \cdot M)/A^b = (g^{ba} \cdot M)/g^{ab} = M$$

Security:

Theorem 11. ElGamal Encryption Scheme is IND-CPA under DDH assumption.

Proof. Suppose that there is a PPT adversary $\mathcal{A}$ that can break the ElGamal Encryption Scheme in the IND-CPA security model with non-negligible advantage ε, we can construct a PPT simulator $\mathcal{B}$ that can solve the DDH problem with non-negligible advantage. Give a DDH problem instance (g, g^a, g^b, T), $\mathcal{B}$ runs $\mathcal{A}$ as the subroutine and works as follows:

- SetUp. $\mathcal{B}$ sets $B = g^a$ and sends it to $\mathcal{A}$.
- Challenge. Upon receiving two different messages M_0, M_1 from $\mathcal{A}$, $\mathcal{B}$ chooses $b \overset{\$}{\leftarrow} \{0, 1\}$ sets the challenge ciphertext $C^* = (g^b, T \cdot M_b)$, and sends C^* to $\mathcal{A}$.
- Guess. $\mathcal{A}$ outputs a guess b' of b. If $b' = b$, $\mathcal{B}$ outputs 1 to indicate that $T = g^{ab}$. Otherwise, $\mathcal{B}$ outputs 0 to indicate that $T = g^c$.

If $T = g^{ab}$, C^* is a well-formed ciphertext under ElGamal Encryption. If $T = g^c$, C^* contains no information of M_b. Thus, the advantage of $\mathcal{B}$ solving the DDH problem is as follows,

$$\begin{aligned}\mathrm{Adv}_{\mathcal{B}} = \Pr[T = g^{ab}]\Pr[b' = b | T = g^{ab}] + \Pr[T = g^c]\Pr[b' \neq b | T = g^c] - \frac{1}{2} \\ = (\varepsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \\ = \frac{\varepsilon}{2}\end{aligned}$$

which is non-negligible. □

6.3 Rivest–Shamir–Adlema Algorithm

Rivest-Shamir-Adleman (RSA) is a widely-used public-key cryptography algorithm, which was developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman [BB79]. It forms the backbone of numerous security applications and protocols, including secure data transmission, digital signatures, and user authentication. The RSA algorithm is based on the mathematical complexity of factoring large integers, particularly those that are products of two large prime numbers. This inherent difficulty ensures the confidentiality and integrity of information exchanged over otherwise insecure channels.

In an RSA system, each participant has a pair of keys: a public key for encryption and a private key for decryption. The public key can be freely shared with others, while the private key must remain confidential to its owner. To encrypt a message, one uses the recipient's public key; only the holder of the corresponding private key can decrypt it. Conversely, if a user wants to digitally sign a document, they would use their own private key, allowing anyone with their public key to verify the authenticity of the signature.

The strength of RSA lies in its asymmetric nature, where the computational ease of performing operations with the keys is highly unbalanced – encryption using the public key is relatively straightforward, but attempting to derive the private key from the public key or ciphertext without proper knowledge is considered computationally infeasible with current technology, especially when sufficiently large key sizes are used. As such, RSA remains a cornerstone of modern cryptography, ensuring the confidentiality, authenticity, and non-repudiation of electronic communications across the globe.

6.3.1 Method

The textbook RSA encryption consists of 3 algorithms:

- KeyGen: Generate primes p and q s.t. $|p| = |q| = \lambda$. Then let $n := p \cdot q$. And randomly select e and d that are co-prime with $\phi(n) = (p-1)(q-1)$ and inverse to each other with modulo $\phi(n)$(that is, $ed \equiv 1(\mod \phi(n))$). Finally output the secret key d and the public key (n, e).

- Enc(e, M): Input the public key e and a message $M \in \mathbb{Z}_n^*$, output the ciphertext $C := M^e$.

- Dec(d, C): Input the secret key d and a ciphertext $M \in \mathbb{Z}_n^*$, output the plaintext message $M' := C^d$.

Note that all multiplication operations are performed within the group $\mathbb{Z}_n^*$.

6.3.2 Correctness and security

Correctness: By Euler's theorem 10 we know that $M' = C^d = M^{ed} = M^{ed \mod \phi(n)} = M$. Hence, the correctness holds.

Security: The IND-CPA security doesn't hold now since the textbook RSA encryption is deterministic, that is, the adversary A can simply pass $\mathbf{Exp}^{\text{ind-cpa}}_{\text{textbool RSA}}$ by output the boolean value $C^* \overset{?}{=} M_1^{*e}$. Now we can only prove it to be a one-way trapdoor permutation directly by RSA assumption, where the trapdoor is d in textbook RSA. To modify the textbook RSA encryption into a IND-CPA scheme, in each encryption we can pad the λ-bit message M with a λ-bit randomness r s.t. $r||M \in \mathbb{Z}_n^*$, and output Enc$(e, r||M)$. Later in decryption we output the last λ bits of Dec(d, C). The random number r ensures that the encrypted results are uniformly distributed among approximately 2^λ numbers but does not affect the correctness of the final decryption. Therefore, the modified RSA encryption is IND-CPA secure and correct.

7 Digital Signature

Digital signatures are a cryptographic mechanism used to verify the authenticity and integrity of a message, software, or digital document. They are akin to a fingerprint that the sender of a message can use to sign their information. The concept of digital signatures is based on public key cryptography, where two keys are used: a private key to create the signature and a public key that others can use to verify it.

Digital signature schemes are built around three fundamental procedures: Key Generation, Signing, and Verification. Here's an overview of each:

- **KeyGen**: Output a pair of keys (sk, vk)—a private key $sk \in \mathcal{SK}$ and a corresponding public key $vk \in \mathcal{VK}$—for the user. the private key is kept secret by the user to create digital signatures. While the public key is shared with others for them to verify signatures made by the private key.

- **Sign**(sk, M): Input the secret key $sk \in \mathcal{SK}$ a message $M\mathcal{M}$, outputs a digital signature $\sigma\sigma$.
- **Vrfy**(vk, M, σ): Input the public key $vk \in \mathcal{VK}$, a message $M\mathcal{M}$, and a signature $\sigma\sigma$, outputs 0/1 to show whether σ is a valid signature of m under the signing key corresponding to vk.

Where $\mathcal{VK}$, $\mathcal{SK}$, $\mathcal{M}$ and $\mathcal{C}$ represents the public key space, the private key space, the plaintext message space and the ciphertext space.

7.1 Comparing to MAC

If we only want to provide integrity of messages, another cryptographic tool, Message Authentication Code(MAC), can also help solve this issue. MAC is a short piece of information used to authenticate a message and to provide integrity and authenticity assurances on the message based on symmetric cryptography, and consisting 3 phases:

- The sender inputs the message and the symmetric key into the MAC algorithm, which outputs a MAC value. This MAC value is based on the message content and the symmetric key and is used to verify the message, not to conceal its content.
- The original message and the generated MAC value are then sent together to the recipient.
- Upon receiving the message and MAC value, the recipient uses the same symmetric key and the same MAC algorithm to process the received message. If the MAC value calculated by the recipient matches the MAC value sent by the sender, the recipient can confirm that the message has not been altered during transmission and that it was indeed sent by the holder of the symmetric key.

The MAC mechanism does have certain weaknesses, owing to its nature as a symmetric encryption technique, and thus the vulnerabilities associated with symmetric encryption also appear in MACs. MACs must ensure that the secret key used by both parties is the same in order to maintain their correctness. Furthermore, if one party's secret key is compromised, then that secret key can no longer guarantee that the message is from a trusted sender, nor ensure the message's integrity.

Compared to MACs, digital signatures based on public key cryptography have the following advantages:

- **Non-repudiation**: Digital signatures provide non-repudiation, which means the sender cannot deny having signed the document. This is because the digital signature is created using the sender's private key, which should be uniquely possessed by the sender.
- **Public Verification**: Digital signatures can be verified by anyone who has access to the public key of the sender. This means that the verification process is not limited to the parties who shared a secret key, as with MACs.

7.2 Defining correctness and security of Digital Signatures

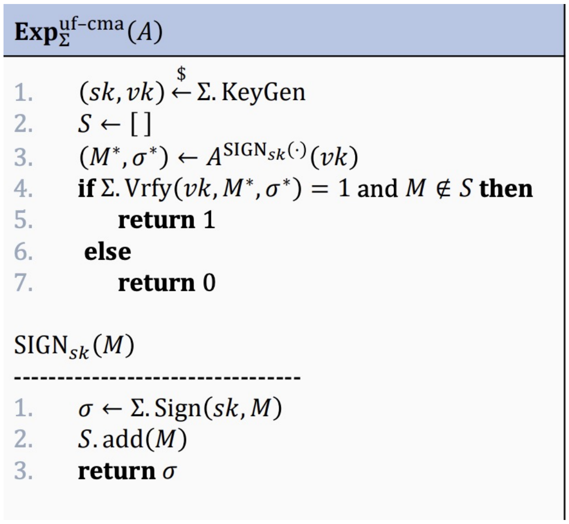
To ensure the correctness of Digital Signatures, we want to guarantee that a message and its corresponding valid message generated by Sign algorithm is valid under the Vrfy algorithm. In a formal description:

Definition 31. Correctness:

$$\forall M \in \mathcal{M}, \Pr[\mathrm{Vrfy}(vk, M, \mathrm{Sign}(sk, M))] = 1$$

where $(sk, vk) \leftarrow$ KeyGen.

Before formally defining security, we need to clarify our security requirements for digital signatures: Adversaries often obtain a collection of messages and their corresponding signatures in advance, and we do not want adversaries to use this information to obtain a signature for a new message. This requirement leads us to the following experiment, where a PPT adversary A holds a vk generated by Σ.KeyGen, and can query a signing oracle SIGN for any message for polynomial times: We wants A to generate a valid signature with only negligible probability. So we define the security like this:



```
$\mathbf{Exp}_{\Sigma}^{\text{uf-cma}}(A)$

$(sk, vk) \overset{\$}{\leftarrow} \Sigma.\mathrm{KeyGen}$
$S \leftarrow [\,]$
$(M^*, \sigma^*) \leftarrow A^{\mathrm{SIGN}_{sk}(\cdot)}(vk)$
if $\Sigma.\mathrm{Vrfy}(vk, M^*, \sigma^*) = 1$ and $M \notin S$ then
    return 1
else
    return 0

$\mathrm{SIGN}_{sk}(M)$
--------------------------------
$\sigma \leftarrow \Sigma.\mathrm{Sign}(sk, M)$
$S.\mathrm{add}(M)$
return $\sigma$
```

Figure 3: UF-CMA experiment

Definition 32. The signature scheme Σ =(KeyGen, Sign, Vrfy) is unforgeable against chosen-message attacks if $\forall$ PPT adversary A, the UF-CMA-advantage (UF-CMA refers to unforgability against chosen-message attacks) of A defined as

$$\mathbf{Adv}_{\Sigma}^{\text{uf-cma}} = \Pr[\mathbf{Exp}_{\Sigma}^{\text{uf-cma}} \Rightarrow 1]$$

is negligible.

7.3 Textbook RSA signatures

The textbook RSA signature scheme involves several steps to generate keys and encrypt/decrypt messages. Below is a detailed description of the process:

1. KeyGen: Nearly the same as the KeyGen for the RSA algorithm. Finally output the private(signing) key as (n, d) and the public(verifying) key as (n, e).

2. Sign$((n, d), M)$: Input the signing key (n, d) and a message $M \in \mathbb{Z}_n^*$, output the signature $\sigma \leftarrow (M^d \mod n)$.

3. Vrfy$((n, e), M, \sigma)$: Input the verifying key (n, e), a message $M \in \mathbb{Z}_n^*$, and the signature σ, output the boolean value $\sigma^e \overset{?}{\equiv} M(\mod n)$.

We can find that given the message-signature pairs (M_1, σ_1) and (M_2, σ_2) under the same secret-public key pair (sk, pk), $\sigma_1\sigma_2$ is a valid signature of M_1M_2 since:

$$(\sigma_1\sigma_2)^d \equiv {\sigma_1}^d{\sigma_2}^d \equiv M_1M_2(\mod n).$$

Thus, the textbook RSA signature scheme is not secure(violating the definition of unforgability against chosen-message attacks).

However, we can modify the Sign and Vrfy algorithms in textbook RSA signature scheme using hash-then-sign. And these two algorithm replace the message M with their hash value $H(M)$, so that the above attack doesn't work by the property of hash that $H(M_1)H(M_2) \not\equiv H(M_1M_2)(\mod n)$ with nearly 1 probability. Moreover, we know the hashed RSA signature scheme is secure if the hash function H is indistinguishable from a perfect random oracle by the following theorem:

Theorem 12. $\forall$ PPT UF-CMA adversary A against hashed RSA making $q\mathsf{SIGN}_{sk}(\cdot)$ queries, there is an PPT algorithm B solving the RSA-problem:

$$\mathbf{Adv}^{\mathsf{uf-cma}}_{\mathsf{RSA},H}(A) \leq q \cdot \mathbf{Adv}^{\mathsf{RSA}}_{n,e}(B)$$

where H is a random oracle.

7.4 Discrete-log-based signatures

Schnorr signatures [Sch91] and ECDSA [JMV01] (Elliptic Curve Digital Signature Algorithm) signatures are cryptographic algorithms used for digital signing and verification. Schnorr signatures are based on the discrete logarithm problem and random oracle model. They were invented by Claus Schnorr and are known for their simplicity and efficiency. ECDSA is a variant of the Digital Signature Algorithm (DSA), which uses elliptic curve cryptography. However, it has no formal security proof and a more complicated design than Schnorr signatures. Yet, it's widely used in many scenarios.

References

[BB79] G Robert Blakley and Itshak Borosh. Rivest-shamir-adleman public key cryptosystems do not always conceal messages. *Computers & mathematics with applications*, 5(3):169–178, 1979.

[Béz79] Etienne Bézout. *Théorie générale des équations algébriques.* Ph.-D. Pierres, 1779.

[BG04] Ian F Blake and Theo Garefalakis. On the complexity of the discrete logarithm and diffie–hellman problems. *Journal of Complexity*, 20(2-3):148–170, 2004.

[Bon98] Dan Boneh. The decision diffie-hellman problem. In *International algorithmic number theory symposium*, pages 48–63. Springer, 1998.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking rsa may not be equivalent to factoring. In *Advances in Cryptology—EUROCRYPT'98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, May 31–June 4, 1998 Proceedings 17*, pages 59–71. Springer, 1998.

[Cay54] Arthur Cayley. Vii. on the theory of groups, as depending on the symbolic equation θn= 1. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 7(42):40–47, 1854.

[CP05] Richard E Crandall and Carl Pomerance. *Prime numbers: a computational perspective*, volume 2. Springer, 2005.

[DH22] Whitfield Diffie and Martin E Hellman. New directions in cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pages 365–390. 2022.

[DK02] Hans Delfs and Helmut Knebl. *Symmetric-Key Encryption*, pages 11–22. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[GSM18] Fuchun Guo, Willy Susilo, and Yi Mu. *Introduction to security reduction.* Springer, 2018.

[JMV01] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1:36–63, 2001.

[Kah96] David Kahn. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet.* Simon and Schuster, 1996.

[Lag71] Joseph-Louis Lagrange. Suite des rflexions sur la résolution algébrique des équations, section troisieme, de la résolution des équations du cinquieme degré & des degrés ultérieurs. *Nouv. Mem. Acad. R. Sci. Berlin*, 202:138–254, 1771.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4:161–174, 1991.

[Sha94] Jeffrey Shallit. Origins of the analysis of the euclidean algorithm. *Historia Mathematica*, 21(4):401–419, 1994.