Lecture note 3: Public key cryptography

Huijiong Yang, Tianyu Zheng, Zhikang Xie

March 15, 2023

In this lecture, we first recall symmetric key cryptography and mention its drawbacks. Next, we introduce a breakthrough work for public key cryptography called Diffie-Hellman Key Exchange. The syntax of public key encryption is presented accordingly, as well as two concrete examples, ElGamal and RSA. At last, we introduce digital signatures as another important application.

1 Recap of symmetric key cryptography

We first introduce a fundamental principle for the design of cryptographic primitives as a supplement to the previous lesson.

Definition 1 (Kerckhoffs' Principle)**.** *Even if attackers have complete knowledge of all the encryption algorithms, the system is secure.*

1.1 Security definitions

As mentioned in lecture 2, the concept of computational security is defined as follows,

Definition 2. *A scheme π is said to be computationally secure if any PPT adversary succeeds in breaking the scheme with negligible probability.*

Specifically, the action of "breaking" is measured by the aim and capability of the adversary described as

- Aim. The aim of the adversary is to try to learn something meaningful from the target ciphertext C^*.
- Capability. For CPA adversary A^{Enc}, it can choose the plaintext and receive the corresponding ciphertext. For CCA adversary $A^{Enc,Dec()}$, it can choose the plaintext or ciphertext and receive the corresponding ciphertext or plaintext.

1.2 Security proof: reduction

If we want to prove that breaking algorithm π is hard, we can use the reduction strategy. There are some hard problems and we assume that problem x is one of them. Then, we just

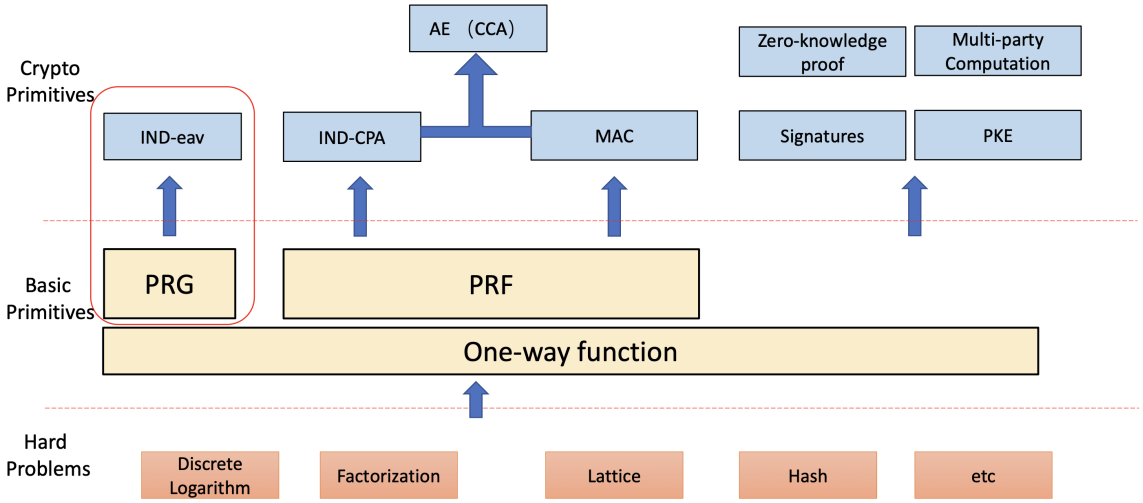

Figure 1: Big picture of Cryptography on page 12, lecture slides 3.

need to prove that if there is an efficient algorithm attacking π, we can use it to solve the problem x, which means that breaking π is harder than x. As a result, if breaking x is hard, we can conclude that algorithm π is secure.

Currently, several hard problems are commonly recognized by the community, which allow us to build lots of useful cryptographic tools with rigorously proven security. The big picture of Cryptography is shown in Figure 1.

1.3 Symmetric-key cryptography

We briefly review the syntax of symmetric-key cryptography. An encryption scheme Π consists of three public algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, standing for the key generation, encryption, and decryption algorithms.

Although symmetric-key cryptography could help us to share messages securely, it also faces some problems. Especially when a user wants to communicate with other $N-1$ users, he has to negotiate and store N symmetric keys. As a result, a system with N users will generate $O(N^2)$ keys in total, which makes it difficult for key management. Some researchers tried to solve this problem more elegantly. Hence the Diffie-Hellman key exchange scheme was proposed.

2 Diffie-Hellman key exchange

Diffie-Hellman key exchange is proposed in 1976 [Hel76], which aims to exchange keys in a secure and efficient way. Generally speaking, Diffie-Hellman key exchange is a two-party protocol for the negotiation of session keys in an insecure channel. Before diving into the details of how it works, we need to introduce some mathematical concepts first.

2.1 Preliminaries

Definition 3. *A group $(G, \circ)$ is a set G together with a binary operation $\circ$ satisfying three axioms:*

- *associativity.* $(a \circ b) \circ c = a \circ (b \circ c)$ *for all* $a, b, c \in G$.
- *identity.* $\exists\ e \in G$ *such that* $e \circ a = a \circ e = a$ *for all* $a \in G$.
- *inverse.* $\forall\ a \in G$ *there exists* $a^{-1} \in G$ *such that* $a \circ a^{-1} = a^{-1} \circ a = e$.

The cyclic group is defined as:

Definition 4. *A group $(G, \circ)$ is cyclic if there exists $g \in G$ such that $G = \{g^i \mid i \in z\} = \{..., g^{-2}, g^{-1}, g^0, g^1, g^2, g^3, ...\}$, where g is a generator for G and $(G, \circ) = \langle g \rangle$.*

For any cyclic group, the following Euler's theorem holds, where e is the identity of the group. Note that this theorem can be proved by contradiction.

Theorem 1. *If $(G, \circ)$ is a finite group, for all $g \in G$, we have $g^{|G|} = e$.*

Corollary 1. $g^i = g^{i \bmod n} = g^{i \bmod |G|}$

Remark 1. *Suppose $p = 2q + 1$ and q is a prime number. $(Z^*{}_p, \cdot)$ has a sub-group $\langle g \rangle$ of order q.*

2.2 Concrete protocols

With the preliminary above, now we can introduce the detailed process of the Diffie-Hellman key exchange. We assume that the two communicating parts are Alice and Bob. Then Alice and Bob execute the following protocol:

Diffie-Hellman key exchange protocol Π_{DH}.

1. Alice and Bob agree on the public parameter (G, p, g), where G is a cyclic group of order p with generator g. // NOTE: (G, p, g) could be the parameter recommended by NIST or RFC document.
2. Alice chooses a random $a \in Z_q$ uniformly and computes $A = g^a$.
3. Alice sends (G, q, g, A) to Bob.
4. Bob receives (G, q, g, A) and chooses $b \in Z_q$. Then, Bob computes and sends $B = g^b$ to Alice. The shared key is computed as $K = A^b$.
5. After receiving B, Alice computes the shared key as $K = B^a$.

Theorem 2. *Diffie-Hellman key exchange Π_{DH} is secure in the presence of an eavesdropper.*

Proof. Suppose that there is a PPT adversary $\mathcal{A}$ that can break the Diffie-Hellman key exchange with non-negligible advantage ϵ, we construct a PPT simulator $\mathcal{B}$ that can break the DDH problem.

Concretely, given as input a problem instance $(G, q, g, A, B, C$ or $K)$ where $(A = g^a, B = g^b, K = g^{ab})$ as instance of DDH problem and C is a random key, $\mathcal{B}$ runs $\mathcal{A}$ and works as follows

- $\mathcal{B}$ sends G, q, g as public parameters of Diffie-Hellman key exchange problem to $\mathcal{A}$.
- $\mathcal{B}$ constructs a Diffie-Hellman key exchange instance as $(A, B, C$ or $K)$ and sends it to $\mathcal{A}$.
- With a non-negligible probability, $\mathcal{A}$ outputs a value b indicating whether the Diffie-Hellman key exchange instance is valid, then $\mathcal{B}$ can distinguish the DDH instance.

□

In order to guarantee the DDH assumption holds, we need to choose a group $\langle g \rangle$ with large space.

3 Syntax of public key encryption

Diffie-Hellman key exchange provides a secure approach for sharing a secret between two parties. As a result, the shared secret can be used as the key for symmetric-key encryption. In this section, we introduce the *public-key encryption*. Different from the former scheme, it directly encrypts the plaintext based on the hard problems.

A public encryption scheme is defined as $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ in the following.

$\mathsf{KeyGen}(1^n)$ On input 1^n, generate $\mathsf{sk} \leftarrow \mathcal{SK}$ and $\mathsf{pk} \leftarrow \mathcal{PK}$, outputs $(\mathsf{sk}, \mathsf{pk})$ as a public and secret key pair.

$\mathsf{Enc}(\mathsf{pk}, M)$. On input public key pk and message M, generate and output $C = \mathsf{Enc}_{\mathsf{pk}}(M)$ as the ciphertext.

$\mathsf{Dec}(\mathsf{sk}, C)$. On input secret key sk and ciphertext C, return $M = \mathsf{Dec}_{\mathsf{sk}}(C)$ as the message or $\perp$ as failure.

The correctness property requires $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)) = M$ holds for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}$, and its security proof is trivial.

Next we introduce the IND-CPA and IND-CCA security of Σ.

3.1 IND-CPA Security

For an adversary $\mathcal{A}$, and security parameter λ, we define the indistinguishably chosen plaintext security of Σ via experiment $\mathsf{Exp}_{\Sigma}^{\mathsf{IND\text{-}CPA}}(\mathcal{A})$.

$\mathsf{Exp}_{\Sigma}^{\mathsf{IND\text{-}CPA}}(\mathcal{A})$

1. The challenger chooses $b \leftarrow \{0,1\}$ to indicate which message is encrypted.
2. The challenger generates $(\mathsf{sk}, \mathsf{pk}) \leftarrow_r \Sigma.\mathsf{KeyGen}$.
3. Adversary $\mathcal{A}$ is given pk, and allowed to find a pair of messages $(M_0^*, M_1^*) \leftarrow \mathcal{A}(pk)$ it wants.
4. Return $\perp$ if $|M_0^*| \neq |M_1^*|$.
5. The challenger runs $C^* = \Sigma.\mathsf{Enc}(\mathsf{pk}, M_b^*)$ and returns back C^*.
6. $\mathcal{A}(\mathsf{pk}, C^*)$ returns b' as the guess of b.
7. Return 1 if $b = b'$, else 0.

Definition 5 (IND-CPA Security)**.** *The* IND-CPA*-advantage of an adversary against* IND-CPA *security of* Σ *is defined as*

$$Adv_{\Sigma}^{ind-cpa}(\mathcal{A}) := |\Pr[Exp_{\Sigma}^{ind-cpa}(\mathcal{A}) \Rightarrow 1] - 1/2|.$$

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

3.2 IND-CCA Security

For an adversary $\mathcal{A}$, and security parameter λ, we define the indistinguishably chosen ciphertext security of Σ via experiment $\mathsf{Exp}_{\Sigma}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$.

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

$$Adv_{\Sigma}^{ind-cca}(\mathcal{A}) := |\Pr[Exp_{\Sigma}^{ind-cca}(\mathcal{A}) \Rightarrow 1] - 1/2|.$$

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

$\mathsf{Exp}_{\Sigma}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$

1. The challenger chooses $b \leftarrow \{0,1\}$ to indicate which message is encrypted.
2. The challenger generates $(\mathsf{sk}, \mathsf{pk}) \leftarrow_r \Sigma.\mathsf{KeyGen}$.
3. Adversary $\mathcal{A}$ is given pk and allowed to access to an oracle $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$. It finds a pair of messages $(M_0^*, M_1^*) \leftarrow \mathcal{A}^{\mathsf{Dec}_{\mathsf{sk}}(\cdot)}(pk)$ it wants.
4. Return $\perp$ if $|M_0^*| \neq |M_1^*|$.
5. The challenger runs $C^* = \Sigma.\mathsf{Enc}(\mathsf{pk}, M_b^*)$ and returns back C^*.
6. $\mathcal{A}^{\mathsf{Dec}_{\mathsf{sk}}(\cdot)}(\mathsf{pk}, C^*)$ returns b' as the guess of b.
7. Return 1 if $b = b'$, else 0.

query $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ with C

1. Return $\mathsf{Dec}(\mathsf{sk}, C)$ if $C \neq C^*$, otherwise abort.

4 ElGamal Encryption

An ElGamal encryption scheme [ElG85] is defined as $\Sigma_{\text{ElGamal}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ in the following.

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$. The key generation algorithm takes as input a security parameter λ, then selects a cyclic group $\mathbb{G}$ according to λ and a corresponding generator g. Next, it selects $x \leftarrow_r \{1, \cdots, |\mathbb{G}| - 1\}$ and returns the public key pk and secret key sk as

$$\mathsf{pk} = g^x, \mathsf{sk} = x.$$

$\mathsf{Enc}(\mathsf{pk}, M) \to C$. The encryption algorithm takes as input a public key pk and a message $M \in \mathbb{G}$. It selects $r \leftarrow_r \{1, \cdots |\mathbb{G}| - 1\}$ and returns the ciphertext C as

$$C = (C_1, C_2) = (\mathsf{pk}^r \cdot M, g^r)\,.$$

$\mathsf{Dec}(\mathsf{sk}, C) \to M$. The decryption algorithm takes as input a secret key sk and a ciphertext $C = (C_1, C_2)$. It returns the message M as

$$M = C_1 / C_2^{\mathsf{sk}}.$$

Theorem 3. *ElGamal Scheme $\Sigma_{ElGamal}$ is correct.*

Proof.

$$\frac{C_1}{C_2^{\mathsf{sk}}} = \frac{\mathsf{pk}^r M}{(g^r)^x} = \frac{(g^x)^r M}{(g^r)^x} = \frac{g^{xr} M}{g^{xr}} = M.$$

□

Theorem 4. *ElGamal Scheme $\Sigma_{ElGamal}$ is IND-CPA under DDH assumption.*

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

- **Setup.** $\mathcal{B}$ sets $\mathsf{pk} = g^a$ and sends it to $\mathcal{A}$.
- **Challenge.** $\mathcal{A}$ outputs two different messages $M_0, M_1 \in \mathbb{G}$. $\mathcal{B}$ chooses $c \leftarrow_r \{0, 1\}$ and sets the challenge ciphertext C^* as

  $$C^* = \left(T \cdot M_c, g^b\right).$$

  Finally, $\mathcal{B}$ sends C^* to $\mathcal{A}$.
- **Guess.** $\mathcal{A}$ outputs a guess c' of c. If $c' = c$, $\mathcal{B}$ outputs 1 to indicate that $T = g^{ab}$. Otherwise, $\mathcal{B}$ outputs 0 to indicate that $T = R$.

If $T = g^{ab}$, C^* is a well-formed ciphertext according to Σ_{ElGamal}. If $T = R$, C^* contains no information about M_c. Thus, the advantage of $\mathcal{B}$ solving the DDH problem is as follows:

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

which is non-negligible. □

5 RSA Encryptions

RSA encryption scheme is another popular scheme proposed by Rivest–Shamir–Adleman in 1977 [BB79]. Different from the previous ElGamal, the RSA scheme is based on another hard problem named RSA problems. We first give an introduction to the basic mathematical theorems it is based on.

5.1 Preliminaries

Lemma 1. *Let a, b be positive integers. $\mathsf{gcd}(a, b)$ can be expressed by $x^*a + y^*b$ where x^*, y^* are integers. Furthermore, $\mathsf{gcd}(a, b)$ is the smallest positive integer that has this formulation.*

Proof. Define a set $I \stackrel{\text{def}}{=} \{xa + yb : x, y \in \mathbb{Z}\}$. Define the smallest positive integer in I as $s \stackrel{\text{def}}{=}_{\min} \{c : c \in I, c > 0\}$. Write $s = x^*a + y^*b$ for some $x^*, y^* \in \mathbb{Z}$. Now we only need to prove that $s = \mathsf{gcd}(a, b)$.

Firstly, we need to prove that $s|a$ and $s|b$. Actually, we can prove this by showing that $s|i$ for every $i \in I$ (obviously $a \in I, b \in I$). Suppose that $i = xa + yb$ for some $x, y \in \mathbb{Z}$. It can be expressed as $i = qs + r$ where $q, r \in \mathbb{Z}$ and $0 \leq r < s$. Then we have

$$r = i - qs = xa + yb - q(x^*a + y^*b) = (x - q \cdot x^*)a + (y - q \cdot y^*)b \in I.$$

If $r \neq 0$, then $0 < r < s$, which contradicts that s is the smallest number in I. Thus, $r = 0$, which proves our claim.

Secondly, we have to show that s is the greatest integer that divides both a and b. Suppose that there is an integer s' such that $s' > s$, $s'|a$ and $s'|b$, then we have $s'|x^*a + y^*b$, namely, $s'|s$, which is a contradiction. Thus, we prove our claim. □

Lemma 2. *Consider the multiplication operation. Let a, N be integers such that $a \geq 1$ and $N \geq 2$. a is invertible modulo N if and only if $\mathsf{gcd}(a, N) = 1$.*

Proof. 1. a is invertible. Suppose b is an inverse of a, we have $ab = 1 \bmod N$. Then we have $ab - 1 = xN$ for some $x \in \mathbb{Z}$ and equivalently $ba - xN = 1$. According to Lemma 1, we have $\mathsf{gcd}(a, N) = 1$ ($b, -x \in \mathbb{Z}$, and 1 is the smallest positive integer).

2. $\mathsf{gcd}(a, N) = 1$. We have $xa + yN = 1$ for some $x, y \in \mathbb{Z}$. Then we have $xa + yN = 1 \bmod N$, namely, $xa = 1 \bmod N$. Thus, x is an inverse of a. □

Theorem 5. *Let N be an integer such that $N \geq 2$, we have $\mathbb{Z}_N^*$ is an abelian group under multiplication modulo N, where $\mathbb{Z}_N^* \stackrel{def}{=} \{a \in \{1, \cdots, N-1\} : \mathsf{gcd}(a, N) = 1\}$.*

Proof. We prove that $\mathbb{Z}_N^*$ satisfies the following axioms.

1. Closure. From Lemma 2, for $a \in \{1, \cdots, N-1\}$, we have that $a \in \mathbb{Z}_N^*$, a is invertible, and $\mathsf{gcd}(a, N) = 1$ are equivalent. For any $a, b \in \mathbb{Z}_N^*$, let $c, d \in \mathbb{Z}$ be inverses of a, b respectively, namely, $ac = 1 \bmod N$ and $bd = 1 \bmod N$. We have $(ab)(dc) = a(bd)c = a1c = ac = 1 \bmod N$, namely, ab is also invertible (bd is an inverse of it). Thus, $ab \in \mathbb{Z}_N^*$.

2. Identity. $1 \in \mathbb{Z}_N^*$.

3. Inverse. Except for using the result of Lemma 2, we also have to find an inverse in $\mathbb{Z}_N^*$ (not only in $\mathbb{Z}$) for any element in $\mathbb{Z}_N^*$. For any $a \in \mathbb{Z}_N^*$, suppose that $b \in \mathbb{Z}$ is an inverse of a. b can be expressed as $b = qN + r$ where $q, r \in \mathbb{Z}$ and $0 < r < N$ (note that $r \neq 0$, otherwise $ab = 0 \bmod N$). We have $ar = a(b - qN) = ab - aqN = 1 - 0 = 1 \bmod N$. From this result: firstly, r is also an inverse of a; secondly, r is invertible and $r \in \{1, \cdots, N-1\}$, namely, $r \in \mathbb{Z}_N^*$.

4. Associativity. This follows from multiplication over integers.

5. Commutativity. This follows from multiplication over integers.

□

Next, we define Euler's Theorem as well as its corollaries.

Theorem 6. *Define $\phi(N) \stackrel{def}{=} |\mathbb{Z}_N^*|$, namely, $\phi(N)$ is the order of the group $\mathbb{Z}_N^*$. For $N = pq$, where p, q are distinct primes, we have $\phi(N) = (p-1)(q-1)$.*

Proof. $N = pq$ have the following positive divisors: $1, p, q, pq$. If an integer $a \in \{1, \cdots, N-1\}$ is not relatively prime to N, namely, $\mathsf{gcd}(a, N) \neq 1$, we have $\mathsf{gcd}(a, N) = p$ or $\mathsf{gcd}(a, N) = q$, equivalently, $p|a$ or $q|a$ (vice versa). Then a can be $p, 2p, \cdots, (q-1)p$ for the first case ($q-1$ elements), or $q, 2q, \cdots, (p-1)q$ for the second case ($p-1$ elements). Thus, we have:

$$\phi(N) = (N-1) - (q-1) - (p-1) = pq - p - q + 1 = (p-1)(q-1).$$

□

Corollary 2. *Let N be an integer such that $N \geq 2$, and $a \in \mathbb{Z}_N^*$, we have*

$$a^{\phi(N)} = 1 \ mod \ N.$$

This can be deduced directly from Theorem 1.

Corollary 3. *Let N be an integer such that $N \geq 2$, for any $a \in \mathbb{Z}_N^*$ and $x \in \mathbb{Z}$, we have*

$$a^x = a^{[x \ mod \ \phi(N)]}.$$

This can be deduced directly from Corollary 1.

5.2 Textbook RSA Encryption

A textbook RSA encryption scheme is defined as $\Sigma_{\text{RSA}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ in the following.

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$. The key generation algorithm takes as input a security parameter λ, then selects two random prime numbers p and q according to λ. Next, it computes $N = pq$ and $\phi(N) = (p-1)(q-1)$. Then, it chooses e from $\mathbb{Z}_{\phi(N)}^*$, namely, $\mathsf{gcd}(e, \phi(N)) = 1$, and computes $d = e^{-1} \bmod \phi(N)$. Finally, it returns the public key pk and secret key sk as

$$\mathsf{pk} = (N, e), \mathsf{sk} = d.$$

$\mathsf{Enc}(\mathsf{pk}, M) \to C$. The encryption algorithm takes as input a public key $\mathsf{pk} = (N, e)$ and a message $M \in \mathbb{Z}_N^*$. It returns the ciphertext C as

$$C = M^e \bmod N.$$

$\mathsf{Dec}(\mathsf{sk}, C) \to M$. The decryption algorithm takes as input a secret key $\mathsf{sk} = d$ and a ciphertext C. It returns the message M as

$$M = C^d \bmod N.$$

Theorem 7. *RSA Scheme Σ_{RSA} satisfies the correctness property.*

Proof.

$$C^d = M^{ed} = M^{[ed \bmod \phi(N)]} = M^1 = M \bmod N.$$

□

The textbook RSA encryption scheme can not be IND-CPA secure since its encryption algorithm is deterministic [KL20]. To achieve IND-CPA or even IND-CCA security, we should pad message with random data in the encryption algorithm. Actually, RSA encryptions are not used commonly in practice.

6 Digital Signature

6.1 Syntax of digital signature

Message authentication code was given in the previous lecture mainly for achieving integrity. Here we introduce a new cryptographic primitive as a digital signature based on public key encryption. Compared with MAC schemes, digital signature has several advantages.

- Public verifiability: Digital signatures can be verified by *anyone*, while MACs can only be verified by a party sharing the same key.
- Non-repudiation: Alice cannot deny having created σ, but she can deny having created a MAC tag T (since Bob could have done it)

A digital signature scheme is defined as $\Sigma_{\text{sig}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vrfy})$ with verification key space $\mathcal{VK}$, and secret key space $\mathcal{SK}$.

$\mathsf{KeyGen}(1^n)$ On input 1^n, generate $\mathsf{sk} \leftarrow \mathcal{SK}$ and $\mathsf{vk} \leftarrow \mathcal{VK}$, outputs $(\mathsf{sk}, \mathsf{vk})$ as a verification and secret key pair.

$\mathsf{Sign}(\mathsf{sk}, M)$. On input secret key sk and message M, generate and output $\sigma = \mathsf{Sign}_{\mathsf{sk}}(M)$ as the ciphertext.

$\mathsf{Vrfy}(\mathsf{vk}, M, \sigma)$. On input verification key vk, message M and signature σ, return $1 = \mathsf{Vrfy}_{\mathsf{vk}}(M, \sigma)$ for valid signature or 0 for invalid one.

The correctness property requires $\mathsf{Vrfy}(\mathsf{vk}, \mathsf{Sign}(\mathsf{sk}, M)) = 1$ holds for all $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}$, and its security proof is trivial.

$\mathsf{Exp}_{\Sigma}^{\mathsf{UF\text{-}CMA}}(\mathcal{A})$

1. The challenger generates $(\mathsf{sk}, \mathsf{vk}) \leftarrow_r \Sigma.\mathsf{KeyGen}$.
2. Adversary $\mathcal{A}$ creates an empty set $S \leftarrow [\,]$.
3. Adversary $\mathcal{A}$ is given vk and allowed to access to an oracle $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$. It finds a pair of messages $(M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}_{\mathsf{sk}}(\cdot)}(vk)$ it wants.
4. The challenger runs $b = \Sigma.\mathsf{Vrfy}(\mathsf{vk}, M^*, \sigma^*)$.
5. Return 1 if $b = 1$, else 0.

query $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$ with M

1. $\sigma \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}, M)$
2. $S.\text{add}(M)$
3. Return σ

Moreover, the security of a digital signature scheme requires that an adversary can not forge a valid signature without knowing the secret key, even if it obtains signatures of many other chosen messages. For an adversary $\mathcal{A}$ and security parameter λ, we define the unforgeable chosen message security of Σ via experiment $\mathsf{Exp}_{\Sigma}^{\mathsf{UF\text{-}CMA}}(\mathcal{A})$.

Definition 7 (UF-CMA Security)**.** *The* UF-CMA*-advantage of an adversary $\mathcal{A}$ against security of Σ is defined as*

$$Adv_{\Sigma}^{uf-cma}(\mathcal{A}) := \Pr[Exp_{\Sigma}^{uf-cma}(\mathcal{A}) \Rightarrow 1].$$

Σ is said to be UF-CMA *secure if for any PPT adversary,* UF-CMA*-advantage is a negligible function of λ.*

6.2 Textbook RSA signatures

A textbook RSA signature scheme is defined as $\Sigma_{RSA} = (\mathsf{KeyGen}_{RSA}, \mathsf{Sign}_{RSA}, \mathsf{Vrfy}_{RSA})$ in the following.

$\mathsf{KeyGen}_{RSA}(1^n)$ On input 1^n, generate two random prime numbers p, q and compute $n = p \cdot q, \phi(n) = (p-1)(q-1)$. Choose e such that $\gcd(e, \phi(n)) = 1$ and $d = e^{-1} \mod \phi(n)$. Output secret key $sk = (n, d)$ and verification key $vk = (n, e)$.

$\mathsf{Sign}_{RSA}(\mathsf{sk}, M)$ On input secret key sk and message M, generate and output $\sigma = M^d \mod n$ as the ciphertext.

$\mathsf{Vrfy}_{RSA}(\mathsf{vk}, M, \sigma)$ On input verification key vk, message M and signature σ, return 1 if $\sigma^e = M \mod n$ or 0 for invalid one.

Theorem 8. *Textbook RSA signature Scheme Σ_{RSA} is correct.*

Proof.

$$\sigma^e = M^{ed} \mod \phi(n) = M^1 = M \mod n$$

□

However, the textbook RSA signature is vulnerable to forgery attacks. Assume that an adversary aims to forge a valid signature with the verification key $\mathsf{vk} = (n, e)$. Then it chooses two different messages and accesses the oracle $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$ for their signatures σ_1, σ_2. As a result, the adversary can forge a valid signature $\sigma = \sigma_1 \cdot \sigma_2 \mod n$ of message $M = M_1 \cdot M_2$ and it can be verified as follows,

$$\sigma^e = (\sigma_1 \cdot \sigma_2)^e \mod n = (M_1 M_2)^{ed} \mod n = M_1 \cdot M_2 = M.$$

6.3 Secure digital signatures

6.3.1 Hash-then sign paradigm

For security and efficiency, we introduce the Hash-then sign paradigm. Instead of signing message m itself, a hash function is used to map it into a fixed-length value first. A hash-and-sign RSA signature scheme is defined as $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vrfy})$ with hash function $H(\cdot)$ in the following.

$\mathsf{KeyGen}_{RSA}(1^n)$ On input 1^n, generate two random prime numbers p, q and compute $n = p \cdot q, \phi(n) = (p-1)(q-1)$. Choose e such that $\gcd(e, \phi(n)) = 1$ and $d = e^{-1} \mod \phi(n)$. Output secret key $sk = (n, d)$ and verification key $vk = (n, e)$.

$\mathsf{Sign}_{RSA}(\mathsf{sk}, M)$ On input secret key sk and message M, generate and output $\sigma = H(M)^d \mod n$ as the ciphertext.

$\mathsf{Vrfy}_{RSA}(\mathsf{vk}, M, \sigma)$ On input verification key vk, message M and signature σ, return 1 if $\sigma^e = H(M) \mod n$ or 0 for invalid one.

Theorem 9. *For any* $\mathsf{UF\text{-}CMA}$ *adversary* $\mathcal{A}$ *against hashed RSA making* q $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$ *queries, there is an algorithm* $\mathcal{B}$ *solving the RSA problem:*

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

where H *is assumed perfect.*

Remark 2. *H is assumed to be a random oracle, which is out of the scope of this course. Refer to [KL20] Section 5.*

6.3.2 Discrete-log-based signatures

Based on the hash-then-sign paradigm above, we further introduce several secure digital signature schemes in real-world applications.

The *schnorr signature* is an elegantly designed scheme with a formal security proof. However, it was patented until February 2008.

The *(EC)DSA signature* is a non-patented alternative scheme derived from the ElGamal-based signature scheme. Compared with the schnorr signature, it has a more complicated design and no security proof. This signature is also standardized by NIST and has a wide range of applications.

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.

[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.

[Hel76] Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

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