ELE2742 / ENG1467 - PUC-Rio, 2019.1

Lecture 6: Public Key Encryption

DATE

Lecturer: Yi-Fan Tseng *Scribe: Yi-Fan Tseng*

Public key encryption is an important primitive in modern cryptography. In contrast with *private key encryption* (or *secret key encryption*), where the encryption and decryption algorithm use the same key, a public key encryption allows us to use *public key* for encryption and *private key* for decryption. The public key can be published, and the private key needs to be kept secret. Public key encryption is also known as *asymmetric encryption*, and private key encryption is also known as *symmetric encryption*.

A public key algorithm consists of the following three algorithms.

$\mathsf{KeyGen}(1^\lambda)$: Taking as input the security parameter 1^λ, this algorithm outputs a pair of public key PK and private key SK.

$\mathsf{Encrypt}(\mathsf{PK}, \mathsf{M})$: Taking as input the public key PK and a message M, this algorithm outputs a ciphertext CT.

$\mathsf{Decyrpt}(\mathsf{PK}\ , \mathsf{SK}, \mathsf{CT})$: Taking as input the public key PK, the private key SK, and a ciphertext CT, this algorithm outputs a message M.

Correctness. A public key encryption is correct if for $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, we have

$$\mathsf{M} = \mathsf{Decyrpt}(\mathsf{PK}, \mathsf{SK}, \mathsf{Encrypt}(\mathsf{PK}, \mathsf{M})).$$

1 RSA Encryption

RSA encryption [4] is proposed by Rivest, Shamir, and Adleman in 1978, which may be the most widely used encryption scheme. The details of the algorithms are shown as follows.

$\mathsf{KeyGen}(1^\lambda)$: Taking as input the security parameter 1^λ, the algorithm performs as follows.

1. Choose two large primes p, q.
2. Compute $N = p \cdot q$.
3. Choose e such that $\gcd(e, \phi(N)) = 1$.
4. Compute $d = e^{-1} \mod \phi(N)$.

5. Output $\mathsf{PK} = (N, e)$ as the public key, $\mathsf{SK} = d$ as the private key.

$\mathsf{Encrypt}(\mathsf{PK}, \mathsf{M})$: Taking as inputs the public key $\mathsf{PK} = (N, e)$ and a message $\mathsf{M} \in \mathbb{Z}_N^*$, the algorithm outputs the ciphertext

$$\mathsf{CT} = \mathsf{M}^e \mod N.$$

$\mathsf{Decyrpt}(\mathsf{PK}, \mathsf{SK}, \mathsf{CT})$: Taking as inputs the public key $\mathsf{PK} = (N, e)$, the private key $\mathsf{SK} = d$, and the ciphertext CT, the algorithm outputs the message

$$\mathsf{M} = (\mathsf{CT})^d \mod N.$$

Correctness. Note that

$$ed = 1 \mod \phi(N),$$

and an element in $\mathbb{Z}_N^*$ has order $\phi(N)$. Thus we have that

$$(\mathsf{CT})^d \pmod N = M^{ed} \pmod N = M^{ed \pmod{\phi(N)}} \pmod N = M \pmod N.$$

We then discuss on the assumption $\mathsf{M} \in \mathbb{Z}_N^*$. If M is not coprime to N, then we have $\gcd(\mathsf{M}, N) = p$ or $\gcd(\mathsf{M}, N) = q$. The probability

$$\mathbf{Pr}[\gcd(\mathsf{M}, N) \neq 1; \mathsf{M} \xleftarrow{\$} [0, N]] = \frac{p+q}{N} = \frac{p+q}{pq}.$$

If $|p| \approx |q|$ and $|N| = 1024$ bits, then the probability

$$\frac{p+q}{N} \approx \frac{2}{\sqrt{N}} \approx \frac{1}{2^{511}},$$

which can be viewed as a negligible term.

2 Rabin Encryption

In 1979, Rabin [3] proposed a variant of RSA encryption, which enjoys the efficient encryption procedure. The detailed algorithms of Rabin encryption are shown below.

$\mathsf{KeyGen}(1^\lambda)$: Taking as input the security parameter 1^λ, the algorithm performs as follows.

1. Choose two large primes p, q.
2. Compute $N = p \cdot q$.
3. Set $e = 2$.
4. Output $\mathsf{PK} = (N, e)$ as the public key, $\mathsf{SK} = (p, q)$ as the private key.

$\mathsf{Encrypt}(\mathsf{PK}, \mathsf{M})$: Taking as inputs the public key $\mathsf{PK} = (N, e = 2)$ and a message $\mathsf{M} \in \mathbb{Z}_N^*$, the algorithm outputs the ciphertext

$$\mathsf{CT} = \mathsf{M}^2 \mod N = \mathsf{M} \cdot \mathsf{M} \mod N.$$

Note that only a modular multiplication is necessary for encryption.
$\mathsf{Decyrpt}(\mathsf{PK}, \mathsf{SK}, \mathsf{CT})$: Taking as inputs the public key $\mathsf{PK} = (N, e)$, the private key $\mathsf{SK} = (p, q)$, and the ciphertext CT, the algorithm outputs the message

$$\mathsf{M} = (\mathsf{CT})^{\frac{1}{2}} \mod N.$$

Note that there will be four square roots of CT. To make the decryption correct, we can pad a pre-determined short string after the message.

3 ElGamal Encryption

The encryption schemes in previous sections are *deterministic*. In a deterministic encryption scheme, the same message will result in the same ciphertext, and thus is not able to withstand the *chosen-plaintext attacks*, where an adversary is allows to obtain the corresponding ciphertexts for any chosen messages. In 1984, Goldwasser and Micali [2] proposed the concept of *probabilistic encryption*, where the same message can be encrypted into different ciphertexts due to the randomness used in the encryption procedure. A concrete instantiation based on quadratic residue are given in the paper.

In this section, we introduce another famous probabilistic encryption scheme, ElGamal encryption, which is proposed by ElGaml [1] in 1985.

$\mathsf{KeyGen}(1^\lambda)$: Taking as input the security parameter 1^λ, the algorithm performs as follows.

1. Choose two large primes p, q, such that $q|p-1$. Let $\mathbb{G}$ be a cyclic group with order q and a generator g.
2. Choose $x \xleftarrow{\$} \mathbb{Z}_q^*$.
3. Compute $y = g^x \mod p$.
4. Output $\mathsf{PK} = (g, y, p, q)$ as the public key, $\mathsf{SK} = x$ as the private key.

$\mathsf{Encrypt}(\mathsf{PK}, \mathsf{M})$: Taking as inputs the public key $\mathsf{PK} = (g, y, p, q)$ and a message $\mathsf{M} \in \mathbb{Z}_N^*$, the algorithm performs as follows

1. Choose $r \xleftarrow{\$} \mathbb{Z}_q^*$.
2. Compute $C_1 = g^r \mod p$.

3. Compute $C_2 = \mathsf{M} \cdot y^r \mod p$.
4. Output $\mathsf{CT} = (C_1, C_2)$.

Decyrpt(PK, SK, CT) : Taking as inputs the public key $\mathsf{PK} = (g, y, p, q)$, the private key $\mathsf{SK} = x$, and the ciphertext $\mathsf{CT} = (C_1, C_2)$, the algorithm outputs the message

$$\mathsf{M} = \frac{C_2}{C_1^x} \mod p.$$

Correctness.

$$\frac{C_2}{C_1^x} = \frac{\mathsf{M} \cdot y^r}{(g^r)^x} = \frac{\mathsf{M} \cdot (g^x)^r}{g^{rx}} = \mathsf{M} \mod p$$

References

[1] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985.

[2] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.

[3] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, USA, 1979.

[4] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120126, Feb. 1978.