CSC 2426: Fundamentals of Cryptography **Fall 2023**

Lecture 9: Public-key Encryption

Instructor: Akshayaram Srinivasan *Scribe: Haohua Tang*

Date: 20 November, 2023

9.1 Public-Key Encryption

We assume Alice and Bob are communication via a public channel, and they have not meet prior to this, so they do not share a secret key. We would like to construct a encryption scheme that works under this scenario, and it should be composed by the following:

$$KeyGen(1^n) \to (pk, sk) \text{ which generates the public and secret key}$$

$$Enc(pk, m) \to \text{Ciphertext } c$$

$$Dec(sk, c) \to \text{Message } m$$

Bob would run *KeyGen* and send the public key to Alice publicly. Alice then would encrypt her message with *Enc* using the public key, and Bob can decrypt them with *Dec* using the secret key.

Now, let us define the correctness and CPA-security of public-key encryption.

Correctness.

$$\forall m, \Pr_{(pk,sk)\leftarrow KeyGen(1^n), c\leftarrow Enc(pk,m)}[Dec(sk, c) = m] = 1$$

CPA-security. We define this by the following security game:

$$Adv \xleftarrow{pk} Challenger, (pk, sk) \leftarrow KeyGen(1^n)$$
$$Adv \xrightarrow[|m_0|=|m_1|]{m_0, m_1} Challenger$$
$$Adv \xleftarrow{c^* \leftarrow Enc(pk, m_b)} Challenger, b \leftarrow 0, 1$$
$$Adv \xrightarrow{b'} Challenger$$

The scheme is CPA-secure if for any computationally bounded adversary

$$\Pr[b' = b] \leq \frac{1}{2} + negl(n)$$

Remark 1. We require $|m_0| = |m_1|$ since if their length is very different (e.g. exponentially), the output of *Enc*, which runs in polynomial time, must also be very different on length and thus trivially distinguishable.

Remark 2. *Enc* must be randomized since otherwise the adversary can just encrypt m_0 and m_1 with the public key and *Enc* (which is also public), thus trivially being able to identify their ciphertext c_0 and c_1.

9.2 Multi-message security

We know that in secret-key encryption one-time security does not imply multi-message security. However this is not the case for public-key encryption.

Claim 9.1 *For public-key encryption, one-time security implies multi-message security.*

Proof: Suppose there exists an adversary A that breaks multi-message security. Let the messages A produces be $\overrightarrow{m_0} = \{m_{0,1}, m_{0,2}, ..., m_{0,p}\}$ and $\overrightarrow{m_1} = \{m_{1,1}, m_{1,2}, ..., m_{1,p}\}$. Then given the random bit b sampled by the challenger, the ciphertext being send to A would be $\{Enc(pk, m_{b,1}), Enc(pk, m_{b,2}), ..., Enc(pk, m_{b,p})\}$, and A distinguishes whether b is 0 or 1 from this with an advantage of $\mu(n)$ which is non-negligible. Now consider the following hybrids:

$$
\begin{aligned}
&H_0 : \{Enc(pk, m_{0,1}), Enc(pk, m_{0,2}), ..., Enc(pk, m_{0,p})\} \\
&H_1 : \{Enc(pk, m_{1,1}), Enc(pk, m_{0,2}), ..., Enc(pk, m_{0,p})\} \\
&H_2 : \{Enc(pk, m_{1,1}), Enc(pk, m_{1,2}), ..., Enc(pk, m_{0,p})\} \\
&\vdots \\
&H_p : \{Enc(pk, m_{1,1}), Enc(pk, m_{1,2}), ..., Enc(pk, m_{1,p})\}
\end{aligned}
$$

By pigeonhole principle, there must exist an $i \in 0, 1, ..., p-1$ such that H_i and H_{i+1} can be distinguished with an advantage of $\frac{\mu(n)}{p}$. With this we construct following adversary B that breaks one-time security. Here we may assume that B knows what i would be since B is in nuPPT.

$$
\begin{aligned}
&A \xleftarrow{pk} B \xleftarrow{pk} Challenger, (pk, sk) \leftarrow KeyGen(1^n) \\
&A \xrightarrow{\overrightarrow{m_0}, \overrightarrow{m_1}} B \xrightarrow{m_{0,i}, m_{1,i}} Challenger \\
&A \xleftarrow[,Enc(pk,m_{1,i+1})...,Enc(pk,m_{1,p})\}]{\{Enc(pk,m_{0,1}),...,Enc(pk,m_{0,i-1}),c} B \xleftarrow{c \leftarrow Enc(pk,m_{b,i})} Challenger, b \leftarrow 0, 1 \\
&A \xrightarrow{b'} B \xrightarrow{b'} Challenger
\end{aligned}
$$

The adversary B would encrypt everything else needed to construct H_i and H_{i+1}, and send them to A, thus using A to distinguish $Enc(pk, m_{b,i})$ by an advantage of $\frac{\mu(n)}{n}$. Hence, we proved the contrapositive of the original claim. ■

9.3 Some Computational Hardness Assumptions

First let us make some computational assumptions on group asthmatics. For a cyclic group $(G, \cdot)$ with order $p \approx exp(n)$ and $g \in G$ being a generator, we assume the following:

- Multiplication, i.e. the $\cdot$ operation can be computed in $poly(n)$ time.
- Given g and any $x \in \mathbb{Z}_p$, g^x can be computed in $poly(n)$ time.

Now we can define the assumption on discrete logarithm.

Proposition 9.2 (Assumption on Discrete Logarithm, DLOG) *For any PPT A,*

$$\Pr_{x \in \mathbb{Z}_p}[A(G, p, g, g^x) = x] \leq negl(n)$$

Remark 3. This directly gives us an one-way function if DLOG is true.

Claim 9.3 *If DLOG is true, we can construct a collision resistance hashing from it.*

Proof: We will construct a hashing that maps $x \in \{0,1\}^m$ to G as follow:

$$Setup(1^n) \to hk = (g^{r_{1,0}}, ..., g^{r_{m,0}}, g^{r_{1,1}}, ..., g^{r_{m,1}})$$

where

$$\forall i, b, r_{i,b} \leftarrow \mathbb{Z}_p$$

and

$$Eval(hk, \mathbf{x}) = \prod_{i=1}^{m} g^{r_{i,x_i}}$$

where $\mathbf{x} = (x_1, ..., x_m)$.

Suppose this is not a CRH. Then there exists an adversary A that given hk sent from a challenger, outputs $\mathbf{x}, \mathbf{x}'$ s.t. with non-negligible probability

$$Eval(hk, \mathbf{x}) = Eval(hk, \mathbf{x}')$$

Now we will construct B such that computes discrete logarithm efficiently.

$$A \xleftarrow[\forall i, b, r_{i,b} \leftarrow \mathbb{Z}_p]{hk'} B \xleftarrow{(G,p,g,g^s)}$$

$$A \xrightarrow[\mathbf{x} \neq \mathbf{x}']{\mathbf{x}, \mathbf{x}'} B$$

where

$$hk' = (g^{r_{1,0}}, ..., g^{r_{i-1,0}}, g^s, g^{r_{i+1,0}}, ..., g^{r_{m,0}}, g^{r_{1,1}}, ..., g^{r_{i-1,1}}, g^{r_{i,1}}, g^{r_{i+1,1}}, ..., g^{r_{m,1}})$$

Since $\mathbf{x} \neq \mathbf{x}'$, there must be a bit that is different, and since B is nuPPT we may assume it is the i-th bit and B knows this will be the case in advance. We also know that with non-negligible probability $Eval(hk', \mathbf{x}) = Eval(hk', \mathbf{x}')$. Thus B computes

$$s' = \sum_j r_{j,1} - \sum_{j \neq i} r_{j,0}$$

and we know with some non-negligible probability $s' = s$. ∎

Now we introduce another assumption. Let the setup for cyclic group be the same with the previous section.

Proposition 9.4 (Decisional Diffie–Hellman Assumption, DDH)

$$\{G, p, g, g^x, g^y, g^{xy}\}_{x,y \leftarrow \mathbb{Z}_p} \approx_c \{G, p, g, g^x, g^y, g^z\}_{x,y,z \leftarrow \mathbb{Z}_p}$$

Remark 4. It is clear that the above proposition implies DLOG.

Claim 9.5 *DDH implies PKE, that is, we can construct a public-key encryption scheme given DDH is true.*

Here is a high-level proof to this claim.

Proof: Let us first construct this PKE scheme.

$$KeyGen(1^n) = (pk, sk) = (g^x, x), x \leftarrow \mathbb{Z}_p$$

$$Enc(pk, m) = c = (c_1, c_2) = (g^r, g^{xr} \cdot m), r \leftarrow \mathbb{Z}_p$$

$$Dec(sk, c) = c_1^{-sk} \cdot c_2$$

Since $Dec(sk, c) = c_1^{-sk} \cdot c_2 = g^{-rx} \cdot g^{xr} \cdot m = m$, we have the correctness. Now we prove security.

Suppose there exists an adversary A that breaks CPA-security of this scheme. For $z \leftarrow \mathbb{Z}_p$, we know that $g^z \cdot m_0 \approx_c g^z \cdot m_1$. However we know that A distinguishes $g^{rx} \cdot m_0$ and $g^{rx} \cdot m_1$ with non-negligible advantage, this it must distinguish at least one of ($g^{xr} \cdot m_0$ and $g^z \cdot m_0$) or ($g^z \cdot m_1$ and $g^{xr} \cdot m_1$) with non-negligible advantage. Hence it distinguishes

$$\{G, p, g, g^r, g^x, g^{rx}\} \text{ and } \{G, p, g, g^r, g^x, g^z\}$$

which contradicts DDH. ■

Remark 5. The above construction has rate=$\frac{1}{2}$, that is, the cyphertext has double the length of the message.

9.4 Trapdoor One-Way Permutation

A trapdoor one-way permutation is composed by the following:

$$Setup(1^n) \rightarrow pk, td \text{ (which stands for } \mathbf{t}\text{rap}\mathbf{d}\text{oor)}$$

$$Eval(pk, x) \rightarrow y$$

where the function $Eval(pk, \cdot)$ is an one-way function and is a permutation. Also it can be inverted given td, that is:

$$Invert(td, y) \rightarrow x$$

Claim 9.6 *Given a trapdoor one-way permutation, we can construct a PKE scheme.*

We provide this construction here. Let h be a hardcore predicate for the one-way permutation $Eval(pk, \cdot)$.

$$KeyGen(1^n) = (pk, td)$$

$$Enc(pk, m) = c = (c_1, c_2) = (Eval(pk, r), h(r) \oplus m), r \leftarrow \mathbb{Z}_p$$

$$Dec(td, c) = h(Invert(td, c_1)) \oplus c_2$$

Remark 5. The above construction has rate=$\frac{1}{n+1}$, that is, to encrypt one bit, it needs to send a ciphertext with length $n + 1$.