CSE 594 : Modern Cryptography 03/28/2017

Lecture 15: Public Key Encryption: I

Instructor: Omkant Pandey *Scribe: Arun Ramachandran, Parkavi Sundaresan*

1 Setting

In Public-key Encryption (PKE), key used for encryption and decryption is different, contrary to Symmetric encryption. Consider when Alice wants to send a private message m to Bob and Alice and Bob do not share any secret. In this case, following are the goals of PKE :

- Public key: Encryption and decryption keys are different. Encryption key is public and decryption key is kept secret.
- Correctness: Alice can compute an encryption c of m using the public key, pk. Bob can decrypt m from c correctly using the secret key, sk.
- Security: The ciphertext of two messages m and $m^{'}$ are computationally indistinguishable. No eavesdropper can distinguish between encryption of m and $m^{'}$ (even using pk)

Unlike Symmetric encryption, the adversary can encrypt a message himself as pk is known to him. So, he does not have to trap the channel. The adversary can simply encrypt a number of known plaintext messages with pk to collect a number of ciphertexts(which are computationally indistinguishable).

2 Definition

- Syntax
  - $Gen(1^n) \rightarrow (pk, sk)$
    polynomial in n, n being the security parameter.
  - $\text{Enc}(pk, m) \rightarrow c$
    Encryption function should be randomized because if it is deterministic, the message will have the same cipher text whenever it is encrypted. The adversary can choose good messages, like 0 and 1, encrypt them(say in a 1-bit encryption scheme), and trap the channel to easily map the pattern(even if sk, is not known)
  - $\text{Dec}(sk, c) \rightarrow m^{'}$ or $\perp$
    Decryption need not be randomized. $\perp$ means the given cipher text is malformed or null, and cannot be decrypted.

  All algorithms run in polynomial time in n.
- Correctness: For every m, $Dec(sk,\ Enc(pk, m)) = m$, where $(pk, sk) \leftarrow Gen(1^n)$

3 Security

IND-CPA means Indistinguishable for Chosen Plaintext attack. It means that the adversary can choose which plaintext to attack. In Symmetric encryption, since the adversary is unaware of the encryption function, in IND-CPA the adversary can get cipher texts for as many messages(whichever ones) as he wants, before he takes up the challenge to distinguish cipher text of two messages.
In PKE, the adversary provides two plain text messages of his choice to test the security(he could either know pk beforehand or not know).

3.1 Weak Indistinguishability Security

The input is the security parameter(n), and pk and sk are generated. The adversary can pick any two messages, m_0 and m_1. The PKE scheme chooses one message at random, encrypts it using pk and gives it to the adversary.

The scheme is weakly indistinguishably secure if the adversary A, when given pk now, is able to tell if the cipher text is an encryption of m_0 or m_1, with probability of no more than $1/2 + \mu(n)$, where $\mu(n)$ is a negligible function. In this case, the adversary was unaware of pk when he picked the two messages m_0 or m_1. In real life, the adversary could know pk beforehand. Even if the adversary were not given pk beforehand, the encryption function could be such that it generates the ciphertext and pk itself.

Definition 1 *A PKE scheme (Gen, Enc, Dec) is weakly indistinguishably secure under chosen plaintext attack (weak IND-CPA) if for all n.u. PPT adversaries A, there exists a negligible function μ such that:*

$$Pr\left[\begin{array}{l}(pk, sk) \xleftarrow{\$} Gen(1^n),\\ (m_0, m_1) \leftarrow A(1^n),: \ A(pk, Enc(pk, m_b)) = b\\ b \xleftarrow{\$} 0, 1\end{array}\right] \leqslant \frac{1}{2} + \mu(n).$$

3.2 Strong Indistinguishability Security

Definition 2 *A PKE scheme (Gen, Enc, Dec) is indistinguishably secure under chosen plaintext attack (IND-CPA) if for all n.u. PPT adversaries A, there exists a negligible function μ such that:*

$$Pr\left[\begin{array}{l}(pk, sk) \xleftarrow{\$} Gen(1^n),\\ (m_0, m_1) \leftarrow A(1^n, pk),: \ A(pk, Enc(pk, m_b)) = b\\ b \xleftarrow{\$} 0, 1\end{array}\right] \leqslant \frac{1}{2} + \mu(n).$$

The second definition is stronger because the scheme is indistinguishably secure even if the adversary was aware of pk and provides messages m_0 and m_1 by taking the information from pk to construct m_0 and/or m_1.

Consider an encryption scheme $Enc(m, pk)$, which is strongly indistinguishably secure. $Enc^{'}(m, pk)$ is defined as below:

$Enc^{'}(m, pk)$
{
  $if(m == pk)$
    $c = pk$
  $else$
    $c = Enc(m, pk)$
}

The above function satisfies weakly indistinguishably secure condition because the probability of m_0 or m_1 being pk, when pk is not known, is negligible. However, it is not strongly indistinguishably secure as the adversary can make pk as $m0$ or $m1$ and guess b correctly. So, IND-CPA is stronger than weak IND-CPA. Indistinguishable security implies weak indistinguishability security.

One-message security implies multi-message security for PKE (by hybrid lemma). Multi-bit messages can be encrypted by picking each bit one-by-one and doing PKE. As mentioned earlier, in symmetric encryption the adversary was not able to encrypt the messages himself while he can now.

4 One Way Functions

Public key encryption scheme for one bit can be built either by using Discrete Log/RSA or by constructing from trapdoor One way permutations(OWP). OWP implies that there is a bijective mapping between image and pre-image. Also, it is a one way function, therefore, it is easy to compute and hard to invert.

A collection of one-way functions is a family $F = \{f_i : D_i \rightarrow R_i\}_{i \in I}$ satisfying the following conditions:

- Sampling function: There exists a PPT Gen such that $Gen(1^n)$ outputs $i \in I$
- Sampling from domain: There exists a PPT algorithm that on input i outputs a uniformly random element of D_i
- Evaluation: There exists a PPT algorithm that on input i, $x \in D_i$ outputs $f_i(x)$.
- Hard to invert: For every non-uniform PPT adversary A, there exists a negligible function $\mu(.)$ such that:

$$Pr\Big[i \leftarrow Gen(1^n),\ x \leftarrow D_i,\ y \leftarrow f_i(x) : f_i(A(1^n, x, y)) = y\Big] \leqslant \mu(n)$$

Theorem 1 *There exists a collection of one-way functions iff there exists a strong one-way function.*

5 Collection of One-way Permutations

A One-way Permutation(OWP) is easy to compute but hard to invert, and bijective.

Definition 3 *A collection $F = \{f_i : D_i \to R_i\}_{i \in I}$ is a collection of one-way permutations if F is a collection of OWFs and for every $i \in I$, f_i is a permutation.*

5.1 Trapdoor Permutations

A trapdoor OWP is such that if we generate and know the secret, it is easy to invert for us but hard for others.

Definition 4 *A collection of trapdoor permutations is a family of permutations $F = \{f_i : Di \to R_i\}_{i \in I}$ that satisfies the following properties :*

- ***Sampling function:*** *$\exists$ a PPT Gen s.t. $Gen(1^n)$ outputs $(i, t) \in I$*
- ***Sampling from domain:*** *$\exists$ a PPT algorithm that on input i outputs a uniformly random element of D_i*
- ***Evaluation:*** *$\exists$ PPT that on input i,$x \in D_i$ outputs $f_i(x)$*
- ***Hard to invert:*** *$\forall$ non-uniform PPT adversary A, $\exists$ a negligible function $\mu(.)$ s.t. :*

$$Pr[i \to Gen(1^n), x \to D_i, y \to f_i(x) : f_i(A(1^n, i, y)) = y] \leq \mu(n)$$

- ***Inversion with trapdoor:*** *$\exists$ a PPT algorithm that given (i, t, y) outputs $f_i^{-1}(y)$*

If we know t, we can invert every image in the range.

$$\forall y \in R_i, Pr[f_i^{-1}(t, y) = x : f_i(x) = y] \; = \; 1$$

5.1.1 Public-Key Encryption from Trapdoor Permutations

Let us consider pk to be f and sk to be t, the trapdoor

$$Enc(pk, m) = f(m)$$
$$Dec(sk, c) = f^{-1}(t, c)$$

The problem with the above approach is that there is no randomization in encryption Enc.
So, let us try to build the encryption scheme for 1-bit (represented below as b)

$$Enc(b) = y = f(r) \quad [a\ PRG]$$

Now, f is known in polynomial time on a random r, and no non-uniform PPT can invert to get r. It is also not possible to tell the hardcore bit of r. Let h be the hardcore predicate (such as the inner product, or something else in case of RSA). Enc will be computed by picking a random r and computing y, and XORing the message m with the hardcore bit of r.

$$Enc(pk, m) : (f(r), h(r) \oplus m)$$

To get the $Dec(sk,(y,c))$, apply sk on y and get r(this will be unique, as it is a permutation), and then XOR $h(r)$ with c

$$Dec(sk,(y,c)) : r \leftarrow f^{-1}(y), output(c \oplus h(r))$$

Representing the above formulation for a family of trapdoor permutations,
$F = \{f_i : D_i \rightarrow R_i\}_{i \in I}$

- $Gen(1^n)$: $(f_i, f_i^{-1}) \leftarrow Gen_T(1^n)$. Output $(pk, sk) \leftarrow ((f_i, h_i), f_i^{-1})$
- $Enc(pk, m)$: Pick $r \leftarrow \{0,1\}^n$. Output $(f_i(r),\ h_i(r) \oplus m)$
- $Dec(sk,(c_1,c_2))$: $r \leftarrow f_i^{-1}(c_1)$. Output $c_2 \oplus h_i(r)$

Theorem 2 *(Gen, Inc, Dec) is IND-CPA secure PKE scheme*

The scheme is secure as the computation of the hardcore bit is secure. The cipher text of m_0 is indistinguishable from that of m_1. The proof is similar to that done in symmetric encryption(using hybrid lemma).

We have to work with a family of trapdoor functions. What does it mean to have a family of trapdoor OWP? We pick i from the names of all the functions in the set. Now, we can pick the name of the function and its domain efficiently. We now look into how to build a family trapdoor OWP below.

5.1.2 Candidate Trapdoor Permutations

RSA $= \{f_i : D_i \rightarrow R_i\}_{i \in I}$ where:

- $I = \{(N,e) | N = p.q \ \ s.t.\ p,q \in \pi_n, e \in Z^*_{\phi(N)} (multiplicative\ group\ with\ inverse)\}$
- $D_i = \{x | x \in Z_N^*\}$
- $R_i = Z_N^*$
- $Gen(1^n) \rightarrow ((N,e),d)\ where\ (N,e) \in I\ and\ e.d = 1\ mod\ \phi(N)\ \ (allows\ to\ build\ the\ trapdoor\ family)$
- $f_{N,e}(x) = x^e\ mod\ N$
- $f^{-1}_{N,d}(y) = y^d\ mod\ N$

To prove that $f_{N,e}(x)$ is a permutation on Z_N^*, we give the inverse, as follows :
If $x \in Z_N^*$, let $f_{N,e}(x) = y = x^e\ mod\ N$

$$\begin{aligned} y^d\ mod\ N &= (x^e)^d\ mod\ N \\ &= x^{de\ mod(\phi(N))}\ mod\ N\ \ (Euler's\ theorem) \\ &= x\ mod\ N\ \ (e.d\ =\ 1\ mod\ \phi(N)) \\ &= x \end{aligned}$$

5.1.2.1 RSA Assumption

For any n.u. PPT adversary A, there exists a negligible function $\mu(.)$ s.t:

$$Pr\Big[\begin{array}{l} p,q \xleftarrow{\$} \pi_n,\ N=p.q,\ e \xleftarrow{\$} Z^*_{\phi(N)}, \\ y \xleftarrow{\$} Z^*_N,\ x \leftarrow\ A(N,e,y) \end{array} : x^e = y\ mod\ N\ \Big] \leq \mu(n)$$

RSA assumption is that without d, it seems hard to compute the e-th root of numbers modulo N. So, even if (N, e) are given as the public key, d and the factors of N could be kept secret, with d being the trapdoor.
The exponentiation operation is costly, so we could just fix a value of e to be a prime number which is relatively prime to $\phi(n)$, such as 3. The encryption becomes fast, but d becomes large, due to which decryption becomes slower. Also, a very small e could enable attacks to break the security of the scheme.

Theorem 3 *Assuming the RSA assumption, the RSA collection is a family of trapdoor permutations.*

6 Food For Thought

There are other security notions of PKE, as follows :

1. Direct and more efficient constructions of PKE, such as by the El-Gamal method
2. - Indistinguishability under chosen-ciphertext attacks(IND-CCA)
   - Circular security/key-dependant message security
   - Leakage-resilient encryption : Even if some part of the secret key is leaked.
3. Weaker security notions such as deterministic encryption which has a limited use but is used for easy indexing and searching in problems