Introduction to Cryptography 2/27/2018

Lecture 13: Public Key Encryption

Instructor: Vipul Goyal *Scribe: Andrew Zigerelli*

1 Hierarchy of Cryptography

We can divide cryptographic primitives and schemes into hierarchies divided by hardness assumptions. Roughly, the existence of primitives in higher floors implies the existence in lower floors.

Floor 0: one time pad

Floor 1: one way functions/permutations $\to$ hardcore predicates $\to$ pseudorandom permutations $\to$ pseudorandom functions $\to$ secret key encrpytion

Floor 2: trapdoor permutations $\to$ public key encryption

2 Public Key Encryption

Definition 1 ***Public Key Encryption Scheme***
A Public Key Encryption scheme (PKE) consists of the following algorithms

- *Gen(n) = (pk, sk)*
- *Enc(pk, m) = c*
- *Dec(sk, c) = m*

satisfying the following

1. *All are PPT.*
2. *Correctness:* $\forall m, \forall (sk, pk) \leftarrow Gen(n), Dec(sk, E(pk, m)) = m$
3. *Indistinguisable Security for PKE as defined below*

Definition 2 ***IND for PKE***
A P.K.E scheme (Gen,Enc, Dec) satisfies ***IND*** *if*
$\forall PPT\mathcal{A}, \forall (m_0, m_1)$,

$$\{(pk, sk) \leftarrow Gen(n) : pk||Enc(pk, m_0)\} \approx_c \{(pk, sk) \leftarrow Gen(n) : pk||Enc(pk, m_1)\} \tag{1}$$

We can also define this in terms for prediction advantage:
$\forall PPT\mathcal{A}, \forall (m_0, m_1)$,

$$\Pr[(pk, sk) \leftarrow Gen(n), b \xleftarrow{\$} \{0, 1\} : \mathcal{A}(pk, Enc(pk, m_b)) = b] \leq \frac{1}{2} + \text{negl}(n) \tag{2}$$

Notice that the above definition holds for all pairs m_0, m_1, no matter who generates them! If we restrict the above defintion to pairs generated by the adversary, this is sometimes called **IND-CPA**; CPA stands for chosen plaintext attack because the adversary (the attacker) is chosing the plaintext (while even knowing the public key), and cannot distinguish between the two produced ciphertexts.

Remark 1 *As with SKE, we must have the encryption not be deterministic. In fact, if deterministic SKE is "bad", then deterministic PKE is a "disaster".*

1. For deterministic SKE, an adversary can notice two ciphertexts are the same. Depending on the situation, this may enable "replay" attacks, in which the an adversary Eve notices that fixed ciphertexts sent from Alice to Bob correspond to some action (e.g. authentication). In this case, the attacker doesn't need to know the plaintext to impersonate Alice; she just needs to send the ciphertexts.

2. For deterministic PKE, an adversary, given a single ciphertext, can encrypt likely messages with the public key and look for the ciphertext. This is devastating if the adversary knows likely messages, or if the size of the message is small. Also, the same "replay" attack still applies if an attacker notices repeating ciphertexts.

Definition 3 ***Multi Message Security for PKE (IND)***
$\forall\{m_0^i\}_{i=1}^{l(n)}, \forall\{m_1^i\}_{i=1}^{l(n)}, l(n) = \text{poly}(n)$

$$\{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_0^i)\}_{i=1}^{l(n)}\} \approx_c \{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_1^i)\}_{i=1}^{l(n)}\} \quad (3)$$

Fortunately, we have a nice theorem that multi message PKE security follows from single message PKE security (under the notion of indistinguishability).

Theorem 1 ***IND for One Time PKE $\implies$ IND for Multi Message PKE***

Proof. Suppose not. Thus, we assume we can distinguish $\{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_0^i)\}_{i=1}^{l(n)}\}$ and $\{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_1^i)\}_{i=1}^{l(n)}\}$, and also that (Gen, Enc, Dec) is One Time PKE Secure (**IND**). We define the following hybrids.

$$
\begin{aligned}
&H_0 : \{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_0^i)\}_{i=1}^{l(n)}\} \\
&H_1 : \{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_1^1), Enc(pk, m_0^i)\}_{i=2}^{l(n)}\} \\
&H_2 : \{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_1^1)_{i=1}^2, Enc(pk, m_0^i)\}_{i=3}^{l(n)}\} \\
&\vdots \\
&H_{j-1} : \{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_1^1)_{i=1}^{j-1}, Enc(pk, m_0^i)\}_{i=j}^{l(n)}\} \\
&H_j : \{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_1^1)_{i=1}^{j}, Enc(pk, m_0^i)\}_{i=j+1}^{l(n)}\} \\
&\vdots \\
&H_{l(n)} : \{(pk, sk) \leftarrow Gen(n) : \{pk, Enc(pk, m_1^i)\}_{i=1}^{l(n)}\}
\end{aligned}
$$

By assumption, H_0 and $H_{l(n)}$ can be distinguished. Thus, by the Hybrid Lmma, $\exists H_{k-1}$ and H_k and a PPT adversary $\mathcal{A}$ s.t. $\mathcal{A}$ can distinguish H_{k-1} and H_k with a noticeable advantage. We use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ to break One Time PKE security, which is a contradiction.

$\mathcal{B}$ works as follows:

1. $\mathcal{B}$ picks (m_0^k, m_1^k) and receives from the oracle $Enc(pk, m_b^k)$.
2. $\mathcal{B}$ invokes $\mathcal{A}$ on the following distribution: $\{\{Enc(pk, m_1^i)\}_{i=1}^{k-1} Enc(pk, m_b^k)\{Enc(pk, m_0^i)\}_{k+1}^{l(n)}\}$
3. $\mathcal{A}$ outputs $b = 0$ (for H_{k-1}) or $b = 1$ (for H_k)
4. $\mathcal{B}$ repeats $\mathcal{A}$'s output

$\mathcal{B}$ is cleary a PPT algorithm as it only creates a polynomial length distribution and simply queries $\mathcal{A}$, which is also PPT. We also remark that the construction of the distribution is such that determining if it's H_k or H_{k-1} exactly determines if m_b is m_1 or m_0. Because PPT $\mathcal{A}$ is correct with noticeable probability, so is $\mathcal{B}$. ■

3 ElGamal PKE

The scheme ElGamal is based on the Decisional Diffie Hellman assumption(**DDH**).

Definition 4 ***Decisional Diffie Hellman (DDH)***
Consider a multiplicative G_q of prime order q and let $g \in G_q$ be a generator. The following distributions are then computationally indistinguishable:

$$\{a, b \xleftarrow{\$} \mathbb{Z}_q : g, g^a, g^b, g^{ab}\} \approx_c \{a, b, r \xleftarrow{\$} \mathbb{Z}_q : g, g^a, g^b, g^{ab}\} \tag{4}$$

Definition 5 ***ElGamal PKE***

1. *Gen(n): Sample $g \leftarrow G, x \leftarrow \mathbb{Z}_q$ Set $h = g^x.pk := (g, h), sk := x$*
2. *Enc(pk,m): Sample $r \xleftarrow{\$} \mathbb{Z}_q$. Output $c = (c_1, c_2) = (g^r, mh^r)$.*
3. *Dec(x,c): Compute c_1^x. Output $c_2(c_1^x)^{-1}$.*

All algorithms are PPT. To show correctness,

$$\begin{aligned} c_2(c_1^x)^{-1} &= mh^r((g^r)^x)^{-1} = mh^r(g^{xr})^{-1} = \\ &= m(g^x)^r(g^{xr})^{-1} = m(g^{xr})(g^{xr})^{-1} = m. \end{aligned}$$

For security, we prove the following lemma.

Lemma 2 $\{g, g^x, g^r, m_0 g^{xr}\} \approx_c \{g, g^x, g^r, m_1 g^{xr}\}$

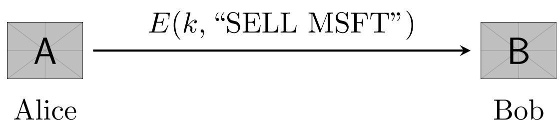
Proof. Sample $R \xleftarrow{\$} \mathbb{Z}_q$. We define the following hybrids:

$$\begin{aligned} &H_0 : \{g, g^x, g^r, m_0 g^{xr}\} \\ &H_1 : \{g, g^x, g^r, m_0 g^R\} \\ &H_2 : \{g, g^x, g^r, g^R\} \\ &H_3 : \{g, g^x, g^r, m_1 g^R\} \\ &H_4 : \{g, g^x, g^r, m_1 g^{xr}\} \end{aligned}$$

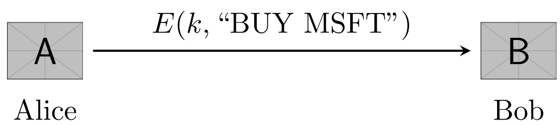
$H_0 \approx_c H_1$ by DDH assumption. $H_1 = H_1 = H_3$ (but we may replace $=$ by $\approx_c$, since equal distributions are obviously computationally indistinguishable.). We can see the equality by viewing multiplication by a fixed element as a permutation on the underlying group. $H_3 \approx H_4$ by DDH assumption. Thus, $H_0 \approx_c H_4$ by transitivity. ■

4 Other Cryptography Concerns

Imagine the following SKE protocol is secure.



Eve, although not shown, can intercept the messages. The following occurs:



What went wrong? The problem is that the schemes described thus far in the class are **malleable**. That is, the attacker can modify the ciphertext with predictable results without knowing the plaintext! We have already seen this for the one time pad. For example, consider Alice sending Bob a single message using a one time pad with previously agreed upon secret key k. Suppose Eve knows that Alice and Bob are sending messages of the form “$CMD||STOCK''$, where “CMD” is 4 bytes (either “SELL” or “BUY␣”) and “STOCK” is a 4 byte stock symbol. Thus, Alice sends $c_1 = (SELL||MSFT) \oplus k$. Eve intercepts c_1, and computes $c_2 = c_1 \oplus (SELL \oplus BUY\text{\textvisiblespace}||0000) = (SELL \oplus BUY\text{\textvisiblespace} \oplus SELL||MSFT \oplus 0000) \oplus k = (BUY\text{\textvisiblespace}||MSFT) \oplus k$. Eve sends c_2 to Bob.

5 ElGamal Attack

Here is an example of how ElGamal is malleable. Suppose Alice is sending a bid, m to Bob, by sending $(c_1, c_2) = (g^r, mh^r)$. Eve wants to bankrupt Alice; thus she intercepts Alice's message and sends $(c_3, c_4) = (c_1, k * mh^r)$ where k is a large positive integer. Upon decryption, Bob will receive a bid of $k * m$, which Alice cannot afford but is now under contract to pay.

6 Non Malleability

For practical value, we need notions of non malleability for encrpytions. Here are informal definitions to be made precise later. For **SKE**, given encryptions of $m_0, m_1, ..., m_n$, an adversary cannot produce an encryption of m_{n+1} s.t. $m_{n+1} \neq m_i \forall 0 \leq i \leq n$.

For **PKE**, given an encrpytion of m, an adversary cannot produce an encrpytion of m' where m and m' are “related”, where “related” will be made precise in the future.