G22.3033-003 Introduction to Cryptography	October 11, 2001
Lecture 6	
Lecturer: Yevgeniy Dodis	*Scribe: Lexing Ying*

1 Public-Key Encryption

Last lecture we studied in great detail the notion of pseudorandom generators (PRG), a deterministic functions that stretch randomness by any *polynomial amount*: from k to $p(k)$ bits. As we already indicated, PRG's have a lot of applications including constructions of both public- and private-key encryptions, and implementation of "ideal randomness" in essentially any programming language. In this lecture we will begin examining these application in more detail by starting with the formal study of public-key encryption (PKE). As we explained before, the informal scenario is this:

- **Before the Encryption.** Alice publishes to the world her public key PK. Therefore, both Bob and Eve know what PK is. This public key is only used to encrypt messages, and a separate key SK is used to decrypt messages. (This is unlike the Secret-Key scheme where one key S is used to both encrypt and decrypt.) Only Alice knows what SK is, and nobody else, not even Bob.
- **Encryption.** When Bob wishes to send Alice a plaintext message M via the Internet, Bob encrypts M using Alice's public key PK to form a ciphertext C. (Formally, we summarize encryption with PK as E_{PK} and say that $C = E_{PK}(M)$.) Bob then sends C over the Internet to Alice.
- **Decryption.** Upon receiving C, Alice uses her secret private key SK to decrypt C, giving her M, the original plaintext message. (Formally, we summarize decryption with SK as D_{SK} and say that $D_{SK}(C) = D_{SK}(E_{PK}(M)) = M$.)
- **Eve's Standpoint** Unlike the Secret-Key scheme, Eve knows everything Bob knows and can send the same messages Bob can. And, only Alice can decrypt. And, when Bob sends his message, Eve only sees C, and knows PK in advance. But, she has no knowledge of SK. And, if it is hard for Eve to learn about SK or plaintexts based on ciphertexts and PK, then our system is secure.

2 Definition of Public-key Encryption

We start with the syntax of a public-key encryptions scheme, and only later talk about its security.

Definition 1 (Public-key encryption (PKE)) *A* PKE *is a triple of* PPT *algorithms* $\mathcal{E} = (G, E, D)$ *where:*

1. *G is the* key-generation algorithm. *$G(1^k)$ outputs (PK, SK, M_k), where SK is the secret key, PK is the public-key, and M_k is the message space associated with the PK/SK-pair. Here k is an integer usually called the* security parameter, *which determines the security level we are seeking for (i.e., everybody is polynomial in k and adversary's "advantage" should be negligible in k).*

2. *E is the* encryption algorithm. *For any $m \in M_k$, E outputs $c \stackrel{r}{\leftarrow} E(m;\ PK)$ — the encryption of m. c is called the* ciphertext. *We sometimes also write $E(m;\ PK)$ as $E_{PK}(m)$, or $E(m;\ r, PK)$ and $E_{PK}(m;\ r)$, when we want to emphasize the randomness r used by E.*

3. *D is the decryption algorithm. $D(c;\ SK) \stackrel{r}{\rightarrow} \tilde{m} \in \{\mathsf{invalid}\} \cup M$ is called the decrypted message. We also sometimes denote $D(c;\ SK)$ as $D_{SK}(c)$, and remark that usually D is deterministic.*

4. *We require the* **correctness** *property: if everybody behaves as assumed*

$$\forall m \in M_k, \quad \tilde{m} = m, \quad \textit{that is} \quad D_{SK}(E_{PK}(m)) = m$$

Example: RSA. Let us check that RSA satisfies the above definition. Notice, both E and D are deterministic.

1. $G(1^k)$ corresponds to the following algorithm: (p, q) are random primes of k bits, $n = pq,\ e \stackrel{r}{\leftarrow} \mathbb{Z}^*_{\varphi(n)},\ d = e^{-1} \bmod \varphi(n),\ M_k = \mathbb{Z}^*_n$. Set $PK = (n, e),\ SK = d$.

2. $c = E(m;\ (n, e)) = m^e \bmod n$.

3. $\tilde{m} = D(c;\ (d, n)) = c^d \bmod n$.

More generally, we could construct a PKE from any TDP. Suppose we have a TDP f with trap-door information t_k and algorithm I for inversion. Here is the induced PKE:

1. $G(1^k) \stackrel{r}{\rightarrow} (f, t_k, \{0,1\}^k)$, and f is the PK and the trapdoor t_k is the SK.

2. $E(m;\ PK) = f(m)$.

3. $D(m;\ SK) = I(c, t_k)$.

Conventions about message spaces M_k. Without loss of generality we will assume that the message space M_k can be determined from the public key PK (so we will not explicitly output its desciption in G). Also, in many schemes the message space M_k does not depend on the particular public key PK and depends only on k, e.g. $M_k = \{0,1\}$ and $M_k = \{0,1\}^k$. In the latter cases we will sometimes say that $\mathcal{E}$ is an encryption for a sequence of message spaces $\{M_k\}$. Notice, however, that sometimes (i.e., in the RSA example above), M_k could depend on the PK.

Problems. The problem of the above general construction is that it does not meet our requirement of security.

- First, it reveals partial information. For example, in the RSA case, $f(m)$ preserves the Jacobi symbol of m. Furthermore, if f' is a TDP $f(a,b) = (a, f'(b))$ is also a TDP however it reveals half of the input message.
- Second, our definition of TDP is based on the assumption of uniform distribution of the input. Here, it corresponds to the uniformity of the distribution on the message space. However, in practice, such uniformity is rarely satisfied, and in some interesting cases, the message spaces is actually quite sparse, for example, the English text.
- Third, when the message space is sparse (i.e., sell/buy), this method is completely insecure and can be broken by a simple exhaustive search.
- Many more problems exist. For example, the adversary can tell whether the same message is being sent twice or not.

For completeness, we would like to point out the general construction does satisfy a very weak security notion.

Definition 2 (One-way secure encryption) *A* PKE $\mathcal{E}$ *is called* one-way secure *if it is hard to* **completely** *decrypt a* **random** *message. Formally, for any* PPT A

$$\Pr(A(c) = m \mid (SK, PK) \xleftarrow{r} G(1^k), m \xleftarrow{r} M_k, c \xleftarrow{r} E_{PK}(m)) \leq \mathsf{negl}(k)$$

Here is a simple lemma directly from the definitions of TDP that shows

Lemma 3 *If* $M_k = \{0,1\}^k$ *is the domain of a* TDP f*, then the* PKE *induced from* f *is one-way secure.*

Conclusions.

- Much stronger definition is needed in order not to reveal partial information.
- Encryption scheme cannot be deterministic in order to solve the problem of non-uniform/sparse message space.
- Even starting with 1-bit encryption, $M_k = \{0,1\}$, is interesting and non-trivial.

3 Secure encryption of one bit

From the previous section, we discussed the problems with one-way security and the straightforward usage of a TDP. In order to have a stronger definition, let us first begin from from trying to encrypt one bit, i.e. $M_k = \{0,1\}$.

One of the conclusions we had is that the encryption scheme must be probabilistic. For each bit from $M_k = \{0,1\}$, there is cloud of messages in the encrypted message space C corresponding to that bit. Informally, we want the distribution of these two clouds to be

indistinguishable to the adversary though their supports are totally disjoint (the disjointness is from the fact that we want to decrypt the message without ambiguity). We write it as $E(0) \approx E(1)$, here $E(0)$ and $E(1)$ are two random variables denoting random encryption of 0 and 1 respectively. Notice this is not possible in the Shannon theory, because there the adversary has infinite power and the only way for the distribution to be indistinguishable is that they are exactly the same, which is not the case since the supports of two distributions are totally different. However, in our case it is doable because we assume the adversary is only PPT. Here is the formal definition.

Definition 4 *A* PKE *for* $M_k = \{0,1\}$ *is called* polynomially indistinguishable *if* $E(0) \approx E(1)$*, meaning that* $\forall$ PPT A

$$\left| \begin{array}{l} \Pr(A(c, PK) = 1 \mid (SK, PK) \xleftarrow{r} G(1^k), c \xleftarrow{r} E_{PK}(0)) - \\ \quad \Pr(A(c, PK) = 1 \mid (SK, PK) \xleftarrow{r} G(1^k), c \xleftarrow{r} E_{PK}(1)) \end{array} \right| \quad \leq \quad \mathsf{negl}(k)$$

Or equivalently,

$$\left| \Pr(A(c, PK) = b \mid (SK, PK) \xleftarrow{r} G(1^k), b \xleftarrow{r} \{0,1\}, c \xleftarrow{r} E_{PK}(b)) - \frac{1}{2} \right| \leq \mathsf{negl}(k)$$

Proof: Let us proof the equivalence claimed in the definition. In the following formulae, we omit the $(SK, PK) \xleftarrow{r} G(1^k)$ part from the conditions for brevity. For the same reason, we omit PK as the input to A. Set

$$\alpha_0 = \Pr(A(c) = 1 \mid c \xleftarrow{r} E_{PK}(0))$$

$$\alpha_1 = \Pr(A(c) = 1 \mid c \xleftarrow{r} E_{PK}(1))$$

Correspondingly,

$$\Pr(A(c) = 0 \mid c \xleftarrow{r} E_{PK}(0)) = 1 - \alpha_0$$
$$\Pr(A(c) = 0 \mid c \xleftarrow{r} E_{PK}(1)) = 1 - \alpha_1$$

The first formula in the definition says that $|\alpha_0 - \alpha_1| \leq \mathsf{negl}(k)$. Since

$$\begin{array}{rcl} \Pr(A(c) = b \mid b \xleftarrow{r} \{0,1\}, c \xleftarrow{r} E_{PK}(b)) & = & \Pr(A(c) = 0 \mid c \xleftarrow{r} E_{PK}(0)) \cdot \Pr(b = 0) + \\ & & \Pr(A(c) = 1 \mid c \xleftarrow{r} E_{PK}(1)) \cdot \Pr(b = 1) \\ & = & \frac{1}{2} \cdot \alpha_1 + \frac{1}{2} \cdot (1 - \alpha_0) \\ & = & \frac{1}{2} + \frac{1}{2} \cdot (\alpha_1 - \alpha_1) \end{array}$$

we get

$$\left| \Pr(A(c) = b \mid b \xleftarrow{r} \{0,1\}, c \xleftarrow{r} E_{PK}(b)) - \frac{1}{2} \right| = \frac{1}{2} \cdot |\alpha_0 - \alpha_1| \leq \frac{1}{2} \cdot \mathsf{negl}(k)$$

Clearly $\frac{1}{2}$ does make a difference to $\mathsf{negl}(k)$, hence these two formulae are equivalent. $\square$

To restate, we can view the definition of indistinguishability as making the following mental experiment. We choose a random bit b and encrypt it. We give the adversary the resulting ciphertext (plus the public key) and ask it to guess b. The scheme is secure if adversary's chances are essentially 1/2. Namely, the best it can do is to flip a coin (independent of c). Put yet another way, the knowledge of c does not help!

Remark 5 *We notice that we will meet the above mental experiment very often. Indeed, whenever we say $X_0 \approx X_1$, we can restate the definition using the same mental game of A having to guess random b based on the sample from X_b. So make sure you really understand the game.*

Example. Suppose f is a TDP with trapdoor information t_k and efficient algorithm I for inversion, and h is a hardcore bit for f. Here is the PKE we informally considered earlier:

1. $G(1^k) \xrightarrow{r} (f, t_k)$.
2. $E(b) \to \langle f(x), h(x) \oplus b \rangle = \langle y, d \rangle$. ($x$ is random in $\{0,1\}^k$).
3. $D(\langle y, d \rangle, t_k)$: $x = I(y, t_k), b = d \oplus h(x)$.

Here is another, slightly more efficient suggestion:

1. $G(1^k) \xrightarrow{r} (f, t_k)$.
2. $E(b)$: sample $x \xleftarrow{r} \{0,1\}^k$ until $h(x) = b$, then set ciphertext $y = f(x)$.
3. $D(y)$: recover $x \xleftarrow{r} I(y, t_k)$, then decrypt $\tilde{b} = h(x)$.

Notice, E is efficient, since sampling the right x will terminate after approximately two trials, since h must be balanced between 0 and 1. We analyze these schemes formally later (or in the homework).

4 Secure Encryption of Many bits

Now, we will consider the case $M_k = \{0,1\}^{p(k)}$, where p is some polynomial in k. The definition is an obvious generalization of the "bit" version.

Definition 6 (Polynomial indistinguishability) *A PKE $\mathcal{E}$ for $M_k = \{0,1\}^{p(k)}$ is called* polynomially indistinguishable *(against PK-only attack) if for any $m_0, m_1 \in M_k$, we have $E(m_0) \approx E(m_1)$. Formally,* $\forall$PPT A

$$\left| \Pr(A(c, PK) = b \mid (SK, PK) \xleftarrow{r} G(1^k), b \xleftarrow{r} \{0,1\}, c \xleftarrow{r} E_{PK}(m_b)) - \frac{1}{2} \right| \leq \mathsf{negl}(k)$$

Comments.

- The definition includes the situation when m_0 and m_1 are the same. In this case, no matter $b = 0$ or $b = 1$, E and A will know nothing about what b is, because they only see the message m_b, which is the same, no matter $b = 0$ or $b = 1$.
- As will will see this definition is extremely robust and prevents a lot of attacks. For example, it also excludes the possibility for the adversary to tell whether a message was being sent twice. Informally, if A could determine this, when given c, A can generate $c' \leftarrow E(m_0)$, and see if c and c' correspond to the same message, thus determining if $b = 0$.

Blum-Goldwasser construction. In the Blum-Goldwasser construction, as we mentioned earlier, we are given a TDP f with trapdoor t_k, inversion algorithm I, and a hardcore bit h. Recall also that if we let $G(x) = G'(x) \circ f^{(n)}(x)$, where $G'(x) = h(x) \circ h(f^1(x)) \circ \cdots \circ h(f^{(n-1)}(x))$, then both G and G' are PRG's (sorry for reusing G same for both key generation and our PRG). We define:

1. $PK = f$ and $SK = t_k$.
2. $E(m)$: get $x \xleftarrow{r} \{0,1\}^k$, send $c = (G'(x) \oplus m, f^{(n)}(x))$.
3. $D(c)$: use t_k to get $f^{(n-1)}(x), \ldots, f(x), x$, and use them to calculate $G'(x)$ with hardcore bit function h. After we have $G'(x)$, m is obvious.

To check the correctness of Blum-Goldwasser construction, we need to prove that $\forall m_0$ and m_1,

$$E(m_0) \equiv (f^{(n)}(x), G'(x) \oplus m_0) \approx (f^{(n)}(x), G'(x) \oplus m_1) \equiv E(m_1) \tag{1}$$

In order to prove this, we will instead prove a more general lemma.

Lemma 7 (One-Time Pad Lemma) *Let R denote the uniform distribution. Then for all distributions X, Y (not necessarily independent!), if $(X, Y) \approx (X, R)$, then for all m_0 and m_1 we have $(X, Y \oplus m_0) \approx (X, Y \oplus m_1)$.*

Proof: Assume not, then we could find a PPT D such that for some ϵ non-negligible,

$$\Pr(D(x, y \oplus m_b) = b \mid (x, y) \xleftarrow{r} (X, Y),\ b \xleftarrow{r} \{0,1\}) \geq \frac{1}{2} + \epsilon$$

We now construct another PPT D', which takes input z and does the following:

$c \xleftarrow{r} \{0,1\}$
$c' \xleftarrow{r} D(x, z \oplus m_c)$
if $c = c'$
  then output 1.
  else output 0.

Intuitively, D' checks if D successfully guessed the right challenge c. If $z \leftarrow Y$,

$$\Pr(D'(x, z) = 1) = \Pr(D(x, z \oplus m_b) = b \mid (x, z) \xleftarrow{r} (X, Y),\ b \xleftarrow{r} \{0,1\}) \geq \frac{1}{2} + \epsilon$$

On the other hand, if $z \leftarrow R$, whether we choose m_0 or m_1 does not make a difference, since z is totally random, so $z \oplus m_c$ is random (and independent of c) as well, so

$$\Pr(D'(x,z)=1) = \Pr(D(R \oplus m_c) = c) = \frac{1}{2}$$

Therefore D' can distinguish between (X,Y) and (X,R), which is contradictory to $(X,Y) \approx (X,R)$. □

We see that Lemma 7 indeed implies the needed Equation (1). Indeed, consider $X = f^{(n)}(x)$, $Y = G'(x)$. A tempting incorrect argument is to say that since G' is a PRG, we have $G'(x) \approx R'$, so

$$(X,Y) \equiv (f^{(n)}(x), G'(x)) \approx (f^{(n)}(x), R') \equiv (X, R')$$

However, this reasoning is wrong! (To see why, replace $f^{(n)}(x)$ by x, and see what goes wrong.[1]) We actually have to use a stronger fact that $G(x) = (f^{(n)}(x), G'(x))$ is a PRG. Now, we can say that $(X,Y) \equiv (f^{(n)}(x), G'(x)) = G(x) \approx R$. Also, $(X,R') \equiv (f^{(n)}(x), R') \equiv R$. The latter follows from the fact that x is random and f is a permutation. Thus, we get

Theorem 8 *BG construction above defines a polynomially indistinguishable encryption.*

As a special case, we also get the security of the one-bit version of BG encryption that we considered in the previous section.

Efficient example: squaring over Blum integers. Recall, the Blum-Blum-Shub construction of G' uses the OWF $SQ(x) = x^2 \bmod n$. This function is a TDP when $n = pq$ with $p = 3 \bmod 4$ and $q = 3 \bmod 4$. Specifically, it can be proved that it is a permutation on SQ_n, and the trapdoor key is the factorization (p,q) of $n = pq$. The associated hardcore bit is the least significant bit of x. Now we see that this construction is quite efficient, because in order to encrypt $p(k)$ bits, we only need to do $p(k)$ multiplications mod n.

5 General transformation from one bit to many bit

Blum-Goldwasser construction shows that given a TDP, we could transfer many bits, by efficiently generalizing the corresponding original BG scheme to encrypt one bit. However, suppose we have another PKE scheme for one bit which possibly does not depend on any TDP. The only thing we know about it is that it is indistinguishable one-bit encryption. Is it possible for us to use this scheme to encrypt many bits without using any other assumptions on this scheme? A naive answer is to regard each bit to be a separate message and encrypt it using the PKE scheme for one bit. Formally, let $\mathcal{E} = (G,E,D)$ be a polynomial indistinguishable PKE scheme for one bit, we could define a PKE scheme $\mathcal{E}' = (G',E',D')$ for $M_k = \{0,1\}^n$ ($n = p(k)$ for some polynomial p) as follows:

[1] More generally, the fact that $A \approx B$ does *not* imply that $(A,C) \approx (B,C)$. Even though this holds if C is *independent* from both A and B (prove this!).

1. $G'(1^k) = G(1^k) \to (PK, SK)$, i.e. PK and SK are generated in the same way as before G, except the message space now is $M_k = \{0,1\}^{p(k)}$. Now, given $m \in M_k = \{0,1\}^n$, we denote m as $m^1 m^2 \cdots m^n$.
2. Define $E'_{PK}(m^1 m^2 \cdots m^n) = (E_{PK}(m^1), E_{PK}(m^2), \cdots, E_{PK}(m^n)) \to c^1 c^2 \cdots c^n = c$.
3. Define $\tilde{m} = D'_{SK}(c^1 c^2 \cdots c^n) = (D_{SK}(c^1), D_{SK}(c^2), \cdots, D_{SK}(c^n))$

It turns out that this bit-by-bit encryption indeed works!

Theorem 9 *If $\mathcal{E}$ is polynomially indistinguishable for one bit, then $\mathcal{E}'$ is polynomial indistinguishable for $n = p(k)$ bits.*

Proof: Take two messages m_0 and m_1. We first construct a sequence of intermediate messages that slowly go from m^0 to m^1:

$$\begin{array}{ccccccccc}
M_0 & = & \boxed{m_0^1} & m_0^2 & \dots & m_0^{n-1} & m_0^n \\
M_1 & = & m_1^1 & \boxed{m_0^2} & \dots & m_0^{n-1} & m_0^n \\
\vdots & & & & \vdots & & \\
M_{n-1} & = & m_1^1 & m_1^2 & \dots & \boxed{m_1^{n-1}} & m_0^n \\
M_n & = & m_1^1 & m_1^2 & \dots & m_1^{n-1} & \boxed{m_1^n}
\end{array}$$

Notice, $M_0 = m^0$ and $M_n = m^n$. Also, M_{i-1} and M_i differ in at most one bit — bit number i. We now define a sequence of distributions

$$C_i \leftarrow E'(M_i) = E(m_1^1) \dots E(m_1^i) E(m_0^{i+1}) \dots E(m_0^n)$$

Using the hybrid argument, in order to prove that

$$E'(m_0) = E'(m_0^1 m_0^2 \cdots m_0^n) \approx E'(m_1^1 m_1^2 \cdots m_1^n) = E'(m_1)$$

i.e. $C_0 \approx C_n$, we only need to show that for any i, we have $C_{i-1} = E'(x_{i-1}) \approx E'(x_i) = C_i$. Graphically,

$$\begin{array}{ccccccccccc}
C_{i-1} & = & E(m_1^1) & \dots & E(m_1^{i-1}) & \boxed{E(m_0^i)} & E(m_0^{i+1}) & \dots & E(m_0^n) \\
C_i & = & E(m_1^1) & \dots & E(m_1^{i-1}) & \boxed{E(m_1^i)} & E(m_0^{i+1}) & \dots & E(m_0^n)
\end{array}$$

So assume that for some i we have $E'(M_{i-1}) \not\approx E'(M_i)$. This is only possible when m_0^i — the i-th bit of M_{i-1}, and m_1^i — the i-th bit of M_i are different, otherwise C_{i-1} and C_i would be exactly the same. Without loss of generality, we denote M_{i-1} and M_i as two strings $s_0 = \alpha 0 \beta$ and $s_1 = \alpha 1 \beta$, where α is a string of $(i-1)$ bits and β is a string of $(n-i)$ bits. Since $E'(s_0) \not\approx E'(s_1)$, then there is a PPT A such that for some non-negligible ϵ (we omit the generation of $(SK, PK) \leftarrow G(1^k)$ for compactness)

$$\left| \Pr(\tilde{b} = b \mid b \xleftarrow{r} \{0,1\},\ c \xleftarrow{r} E'_{PK}(s_b),\ \tilde{b} \xleftarrow{r} A(c,\ PK)) - \frac{1}{2} \right| \geq \epsilon$$

or since $s_b = \alpha b \beta$,

$$\left| \Pr(\tilde{b} = b \mid b \xleftarrow{r} \{0,1\},\ \tilde{c} \xleftarrow{r} E_{PK}(b),\ \tilde{b} \xleftarrow{r} A(E'_{PK}(\alpha) \circ \tilde{c} \circ E'_{PK}(\beta),\ PK)) - \frac{1}{2} \right| \geq \epsilon$$

Now we almost immediately see how to construct a PPT A' which takes $\tilde{c}$, an encryption of unknown message $b \in \{0,1\}$, and the public key PK as input, and tries to guess this b:

set $c = (E'_{PK}(\alpha) \circ \tilde{c} \circ E'_{PK}(\beta))$.
output $\tilde{b} = A(c, PK)$.

It is now clear that the advantage of A' is at least ϵ. Here we used the fact that A' can compute $E'(\alpha)$ and $E'(\beta)$ without any knowledge of b, by simply using the public key PK. Thus, we contradict the fact that E is polynomially indistinguishable for bits. □

To recap the whole proof, we used the fact that if one can distinguish between encryptions of two long messages encrypted bit-by-bit, there must be some particular index i that gives the adversary this advantage, but this contradicts the bit security of our base encryption scheme. Also, notice that we loose a polynomial factor $p(k) = n$ in security by using the hybrid argument, but this is OK since n is polynomial.

Remark 10 *We notice that the above one-bit to many-bit result is false for private-key encryption (which we did not cover formally yet). For example, consider the one-time pad with secret bit s and $E_s(b) = b \oplus s$. We know it is perfectly secure. However, if we are to encrypt two bits using s, $c_1 = b_1 \oplus s$ and $c_2 = b_2 \oplus s$, then $c_1 \oplus c_2 = b_1 \oplus b_2$, so we leak information. The key feauture of the public-key encryption that makes the result true is the fact that anyone can encrypt using the public key, which is false in the private-key case (check that this is the place where the proof fails).*