6.875J/18.425J Cryptography and Cryptanalysis	September 24, 2001
Lecture 5	
Instructor: Anna Lysyanskaya	*Scribed by: Stephen Weis*

1 Recap

1.1 The Definition of Security for Binary Message Spaces

A cryptosystem consists of three algorithms, G, E, D. $(PK, SK) \leftarrow G(1^k)$, $c \leftarrow E(PK, m)$, $m = D(SK, c)$. Such a cryptosystem is considered secure if:

$$\begin{aligned}
&\forall PPTF\{Eve_k\}\,\exists f(k) \in \text{negl}\,\exists K \text{ such that for } k > K:\\
&\quad \Pr[\ (PK, SK) \leftarrow G(1^k);\\
&\quad\quad b \leftarrow \{0,1\};\\
&\quad\quad c \leftarrow E(PK, b);\\
&\quad\quad b' \leftarrow \text{Eve}(PK, c) \quad : \quad b = b'] \leq 1/2 + f(k)
\end{aligned}$$

Where negl denotes the set of negligible functions,[1] and $PPTF$ stands for "probabilistic polynomial time family" of Turing machines.

1.2 The Quadratic Residuosity Assumption

Let $RSAM_k$ be the algorithm for selecting an RSA modulus with primes of length k. Let $J_1(n)$ denote the set of elements of $\mathbb{Z}_n^*$ mod n with Jacobi Symbol 1. Let $QC(a, n)$ denote the quadratic character of a modulo n. That is to say, $QC(a, n) =$ "square" if a is a square and $QC(a, n) =$ "non-square" otherwise. Then, the quadratic residuosity assumption is as follows:

$$\begin{aligned}
&\forall PPTF\{A_k\}\,\exists f(k) \in \text{negl}\\
&\quad \Pr[\ n \leftarrow RSAM_k;\\
&\quad\quad a \leftarrow J_1(n);\\
&\quad\quad b' \leftarrow A_k(a, n) \quad : \quad b' = QC(a, n)\ ] \leq 1/2 + f(k)
\end{aligned}$$

Let $QR(n)$ denote the set of quadratic residues modulo n. Let $QNR(n)$ denote the set of quadratic non-residues modulo n with Jacobi symbol 1. Note that $a \leftarrow (QR(n) \cup QNR(n))$ is equivalent to $a \leftarrow J_1(n)$.

An equivalent way of writing the selection of a is: $(d \leftarrow \{0,1\}; a_0 \leftarrow QR(n); a_1 \leftarrow QNR(n); a \leftarrow a_d)$. That is to say, we first choose d, the quadratic character of a, and then choose a in accordance with d.

So the quadratic residuosity assumption can also be written as follows:

$$\begin{aligned}
&\forall PPTF\{A_k\}\,\exists f(k) \in \text{negl}\\
&\quad \Pr[\ n \leftarrow RSAM_k;
\end{aligned}$$

[1] Recall that $f(k) \in$ negl if for all $c \in \mathbb{N}$, there exists a K such that for all $k \geq K$, $f(k) < 1/k^c$.

$d \leftarrow \{0,1\};$
$a_0 \leftarrow QR(n);$
$a_1 \leftarrow QNR(n);$
$d' \leftarrow A_k(a,n) \quad : \quad d' = QC(a,n) \ ] \leq 1/2 + f(k)$

2 The Goldwasser-Micali Cryptosystem

The GM cryptosystem is as follows:

Key Generation G Alice chooses two random k-bit primes, p and q. Let $n = pq$. Alice also chooses a random $x \in QNR(n)$. I.e., x is a random non-square with Jacobi symbol 1. Alice's public key PK is the tuple (n, x), while her secret key SK is n's factorization (p, q).

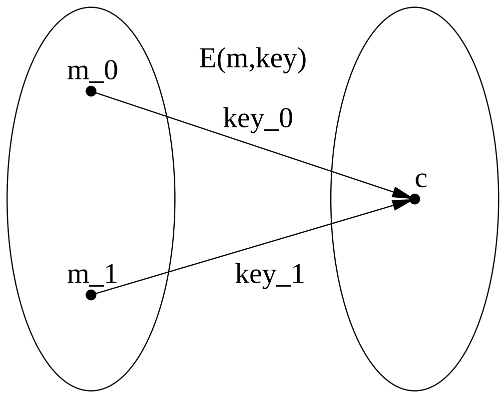
Encryption E To encrypt a 0, Bob sends Alice a random square modulo n. To encrypt a 1, Bob sends Alice a random non-square with Jacobi symbol 1. More precisely, $c \leftarrow E(n, x, b)$ is computed as follows: $(r \leftarrow \mathbb{Z}_n^*; c \leftarrow r^2 x^b \bmod n)$.

Decryption D To decrypt, Alice determines whether her received value c is a quadratic residue modulo n. More precisely, Alice tests whether $\left(\frac{c}{p}\right) = \left(\frac{c}{q}\right) = 1$. In Lecture 3, we saw an efficient algorithm for doing this.

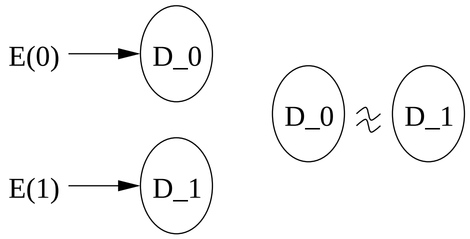
3 Definition of GM-security for l-bit messages

3.1 Discussion on indistinguishability

What was our first attempt at defining security, in lecture 1? It was that for all messages m_0, m_1, the distribution induced by encrypting m_0 is the same as the distribution induced by encrypting m_1.



Now that we have a definition for binary message spaces, it rests on the intuition that, although the distributions induced by 0 or 1 have to be distinct, no adversary should be able to tell them apart, i.e., they are *indistinguishable*.

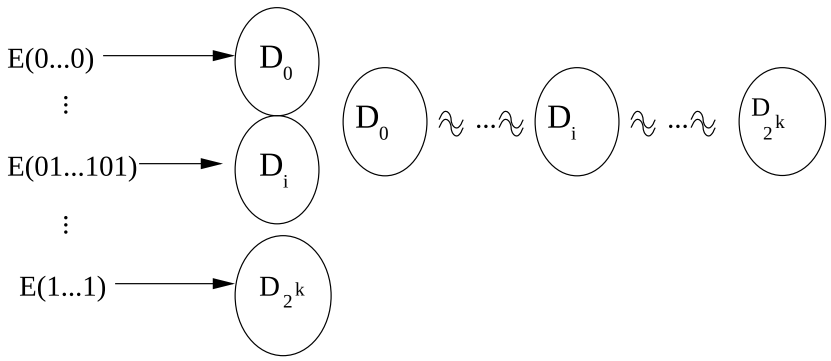


A remark: indistinguishability means that for a *random* sample selected from one of the distributions, we cannot tell whether it was selected from D_0 or D_1. It does *not* mean that for *any* element, the distribution it came from cannot be determined. For example, 4 is always a square, but this does not contradict the quadratic residuosity assumption.

Exercise: using the notation we developed, formulate the statement "Distributions D_0 and D_1 are indistinguishable."

3.2 Definition

Now let us extend this intuition to larger message spaces, as follows: for all m_0, m_1, the resulting distributions are indistinguishable.



More formally:

Definition 1 *A cryptosystem* (G, E, D) *for message space* $\{0,1\}^l$ *is* GM-secure *if*

$$\begin{aligned}
&\forall PPTF\{Eve_k\}\forall m_0, m_1 \in \{0,1\}^l \exists f(k) \in negl \exists K \text{ such that for } k > K:\\
&\quad \Pr[\ (PK, SK) \leftarrow G(1^k);\\
&\qquad b \leftarrow \{0,1\};\\
&\qquad c \leftarrow E(PK, m_b);\\
&\qquad b' \leftarrow Eve(PK, c) \quad : \quad b = b'] \leq 1/2 + f(k)
\end{aligned}$$

This definition was first given my Shafi Goldwasser and Silvio Micali and it is usually referred to as "GM-security," where GM can be de-abbreviated both as "Goldwasser-Micali" and "general message." (When Silvio teaches this course, he always uses the latter de-abbreviation because he is modest. But we prefer the former because it is more true.)

4 GM cryptosystem is secure for l-bit messages

Suppose we need to encrypt an l-bit message m, but we only have an encryption algorithm for one bit. The first thing that comes to mind is to encrypt m bit-by-bit. This is exactly what Goldwasser-Micali propose:

Key Generation G : same as in the binary message case: the public key consists of an RSA modulus n and a quadratic residue a. The secret key is the factorization of n.

Encryption E : Break up the message into bits, denoted $m = b_1 \circ b_2 \circ \ldots \circ b_l$. Encrypt them one-by-one: $c = r_1^2 a^{b_1}, \ldots, r_l^2 a^{b_l}$.

(We need to choose a new random value, r_i otherwise Eve would see the same two ciphertexts, c_0 and c_1, being repeated corresponding to the bit values of the plaintext message. This will enable Eve will to gain a lot of information about the message sent.)

Decryption D : same as in the binary message case, for each bit.

How do we prove security? Let us look at the original reduction one more time: on input (n, a), set $PK = (n, a)$. Then, choose b,r at random and let $c = r^2 a^b \bmod n$. Obtain $b' = \text{Eve}(PK, c)$. If $b' = b$, output "Non-square" (or 1) Otherwise, output "Square" (or 0).

Now let us extend it to the case with many bits: same as before, on input (n, a), set $PK = (n, a)$. Then, choose d,$r_1, \ldots, r_l$ at random and let $c_i = r_i^2 a^{b_i^d} \bmod n$, $1 \leq i \leq l$. Obtain $d' = \text{Eve}(PK, c)$. If $d' = d$, output "Non-square" (or 1) Otherwise, output "Square" (or 0).

Suppose a is a quadratic residue. Then, no matter what message is encrypted, the ciphertext c will be just a bunch of quadratic residues. This is because if $a = g^2$ for some g, then $E(b) = r^2 a^b = r^2 (g^2)^b \bmod p$. This message could just as likely be an encryption of 0 where $r_0 = rg$, so the probability that Eve will guess d correctly for any bit is exactly $1/2$, as before.

Now, suppose that a is a non-residue. Then Eve's probability of success is as in the actual attack. Therefore, this reduction goes through exactly the same as in the one-bit case, and the GM cryptosystem is secure for longer messages as well.

This was a high-level argument — but I don't want to make it formal and precise, because it is not worth it! Instead, let's prove something much more general!

5 GM-Security for bits implies GM-security for strings

In fact, it is the case that any cryptosystem that is GM-secure for binary messages is GM-secure for messages of arbitrary length.

Theorem 2 *Suppose a cryptosystem (G', E', D') is GM-secure for one-bit messages. Then the following cryptosystem (G, E, D) is secure for l-bit messages:*

Key Generation G *: Same as G'.*

Encryption E *: Use E' for each bit of m, individually.*

Decryption D : *Using D', decrypt each individual bit.*

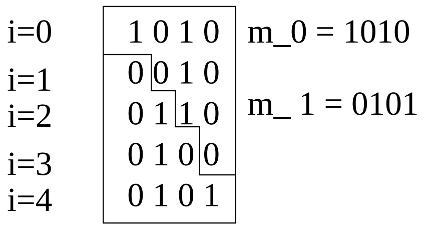
Proof Suppose, for contradiction, that (G, E, D) is not secure. We wish to show that this implies that (G', E', D') is not secure either.

What does it mean that (G, E, D) is not GM-secure? It means that there exists an adversary $\{A_k\}$ such that for infinitely many k, the exist messages m_0 and m_1 and polynomial $p(k)$ such that

$$\begin{aligned}&\Pr[(PK, SK) \leftarrow G(1^k); b \leftarrow \{0,1\};\\ &\quad c \leftarrow E(PK, m_b); b' \leftarrow A_k(PK, c) \quad : \quad b = b'] \geq 1/2 + 1/p(k)\end{aligned}$$

Now let us define the $l+1$ *hybrid* messages as follows: for $0 \leq i \leq l$, m_k^i is obtained by taking the first i bits of m_1 and concatenating them with the last $l-i$ bits of m_0.

For example, suppose $m_0 = 1010$, and $m_1 = 0101$. Then the hybrids are as follows:



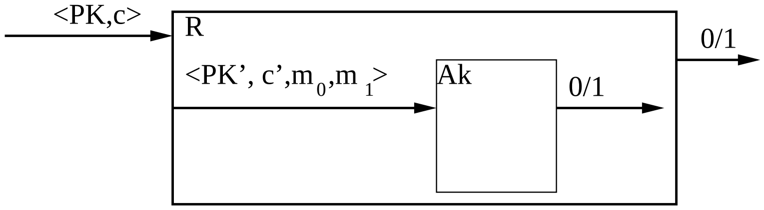
Claim 3 *For some $0 \leq i < l$,*

$$\begin{aligned}&\Pr[(PK, SK) \leftarrow G(1^k); b \leftarrow \{0,1\};\\ &c \leftarrow E(PK, m^{i+b}); b' \leftarrow A_k(PK, c) \quad : \quad b = b'] \geq 1/2 + 1/(l * p(k))\end{aligned}$$

Moreover, given A_k, m_0 and m_1, one can determine i.

Let us take the claim for granted for now, and use it in the following reduction. First let us make the following observations: (1) the messages m_0 and m_1 must differ in their $(i+1)$st bit, because otherwise m^i and m^{i+1} would be the same message and so no adversary would be able to tell their encryptions apart; (2) without loss of generality, we can assume that the $(i+1)$st bit of m_0 is 0, but the $(i+1)$st bit of m_1 is 1: they can be re-labelled if this is not the case.

Our reduction R receives as input a public key PK' from the cryptosystem (G', E', D') and a challenge ciphertext c'. Our goal is to determine whether c is an encryption of 0 or an encryption of 1.



Now we must create the challenge ciphertext for our adversary A_k, as follows: the first i blocks of c will be encryptions of the first i bits of m_1. The $i+1$st block will be the input ciphertext c'. The last $l-i-1$ blocks will be the encryptions of the last $l-i-1$ bits of m_0. Note that, in case c' is the encryption of 0, c will be the encryption of m^i; otherwise c will be the encryption of m^{i+1}. Our reduction will simply output whatever A_k outputs. Let us estimate the success probability of our reduction R_k.

Let q denote the probability that R_k's answer is correct. First, note that

$$\begin{aligned}
q = \Pr[(PK', SK') \leftarrow G';\\
b \leftarrow \{0,1\}; c' \leftarrow E'(PK', b);\\
b' \leftarrow R_k(PK', c') \quad &: \quad b = b'] =\\
\Pr[(PK', SK') \leftarrow G';\\
b \leftarrow \{0,1\}; c' \leftarrow E'(PK', b);\\
c \leftarrow (E'(PK', m_{k,1,1}), \ldots, E'(PK', m_{k,1,i}), c', E'(PK', m_{k,0,i+2}), \ldots, E'(PK', m_{k,0,l}));\\
b' \leftarrow A_k(PK', c) \quad &: \quad b = b']
\end{aligned}$$

This is because of the way we have constructed the reduction: it outputs precisely the same value b' as A_k.

Next, note that in case c' is the encryption of 0, c will be the encryption of m^i; otherwise c will be the encryption of m^{i+1}. Therefore, our reduction will be correct precisely in the cases when A_k correctly tells between the two hybrids:

$$q = \Pr[(PK, SK) \leftarrow G; b \leftarrow \{0,1\}; c \leftarrow E(m_k^{i+b}); b' \leftarrow A_k(PK, c) : b = b']$$

Finally, by Claim 3, we get that $q \geq 1/2 + 1/(l * p(k))$.

We will postpone the proof of Claim 3 until Lecture 6. ■

The proof technique we used is called *the hybrid argument.* Its idea is to look at the distributions "in between" the two distributions that we are interested in, and show that if our input distributions are distinguishable, then some "neighbor" distributions are distinguishable too.