Lecture Notes 13

Scribe: Anthony Lobay - lobay@cs

1 Quick Intro

Last time we introduced the notion of a hard core bit (HCB). The basic idea now, is that given a hard core predicate B, a 1-bit message m can be made secure in the public key sense.

Specifically, given the Goldwasser-Micali (GM) encryption scheme,
$E_{pk}(m) = (F_pk(r), B_{pk}(r) \oplus m)$
we can say:
$(t, \epsilon) - HCB \Rightarrow (t, q, q\epsilon) - IND - CPA$
for our one bit message m.

2 Hard-Core Bits

We say that when a precicate B is $(t, \epsilon) - hard - core$ for F:
$AdvA = 2|\mathbf{Pr}_{(pk,tk)\leftarrow G}[A(pk, F_{pk}(x)) = B_{pk}(x)] - \frac{1}{2}| \leq \epsilon \ \forall A$
*See previous class for definitions of HCBs.

Example 1 (LSB for RSA) For RSA, the least-significant bit (LSB) of x has been shown to be a HCB (making the RSA Assumption). The predicate is simply $B(x) = LSB(x)$ for $F_n(x) = x^3 \bmod n$ [ACGA]

Student Question: Would this imply RSA is as strong as factoring? Answer: So... it looks like they're relying on making the RSA assumption.

How would one encrypt 1 bit b securely under the RSA assumption? Create the following scheme: $E_{pk}(b) = (x^3 \bmod n, LSB(x) \oplus b)$ where x is chosen at random ($x \in (\mathbb{Z}|n\mathbb{Z})^*$). Now given this scheme, who can break it? Who can decrypt with the trap-door? That's what we're after.

Let's make a claim about the security of the above scheme.

3 Encryption of 1 Bit

We'd like to show the scheme:

$E_{pk}(b) = (x^3 \bmod n, LSB(x) \oplus b)$ where x is chosen at random, is secure in the LR-semantic sense.

THEOREM 1
If the RSA-Assumption is true, the above scheme is IND-CPA secure.

Proof: If you, Eve, could predict b, you could predict $x^3 \bmod n$. This, of course, is impossible. Why? We're actually going to use a message m larger than 1 bit with RSA. The result is that to predict m Eve would have to predict $x^3 \bmod n$, which contradicts the RSA assumption.

So the above scheme is IND-CPA secure.

This should suggest a way to get a 1-way permutation. Suppose B is $(t, \epsilon) - hard - core$ for F. Then we can create the scheme $E_{pk}(m) = (F_{pk}(x), B_{pk}(x) \oplus m)$ where x is chosen uniformly at a random. This turns out to be the Goldwasser-Michali encryption scheme.

Student Comment: I don't see how this fits with the RSA-Assumption. Response: If I pick x^3, you can't guess x.

THEOREM 2 (GOLDWASSER MICALI)
E is $(t, 1, \epsilon) - LR - secure$

As usual, we'll prove the contrapositive followed by a reduction. Suppose that that E is not secure in the above sense. Then we can write:

$$\mathbf{Pr}_{m \in \{0,1\}}[A(pk, E_pk(m)) = m] > \frac{1}{2} + \frac{\epsilon}{2}$$

The attacker queries messages 0 and 1 and tries to distinguish between the two. Here m is a bit drawn uniformly at random. (There's no advantage is using anything more than {0,1}.) This means:

$$\mathbf{Pr}_{m \in \{0,1\}, x}[A(pk, (F_{pk}(x), \underbrace{B(x) \oplus m}_{z})) = m] > \frac{1}{2} + \frac{\epsilon}{2}$$

Make a simple change of variables.

$$\mathbf{Pr}_{z \in \{0,1\}, x}[A(pk, (F_{pk}(x), z)) = z \oplus B_{pk}(x)] > \frac{1}{2} + \frac{\epsilon}{2}$$

This means one can construct a natural algorithm A' that we'll use to break B.
Define $A'(pk, y)$:

1. *Pick $z \in \{0, 1\}$*
2. *Output $A(pk, (y, z)) \oplus z$*

Thus:

$$\mathbf{Pr}_x[A'(pk, F_{pk}(x)) = B_{pk}(x)] = \mathbf{Pr}_{x,z}[A(pk, (F_{pk}(x), z)) \oplus z = B_{pk}(x)]$$
$$= \mathbf{Pr}_{x,z}[A(pk, (F_{pk}(x), z)) = z \oplus B_{pk}(x)] > \frac{1}{2} + \frac{\epsilon}{2}$$

So if you can break E w/ Adv ϵ, then you can break B with Adv ϵ. This of course contradicts the fact that $B$is $(t, \epsilon) - hard - core$.

Corollary 3 (E is $(t, q, q\epsilon) - LR - secure$)
We'll make a (quick) hybrid argument. Given we know that E is $(t, 1, \epsilon) - LR - secure$ we can make use of the triangle inequality. For each query, we gain advantage of at most ϵ. By the triangle inequality, we know that if for a gain of ϵ per query and q queries, E is $(t, q, q\epsilon) - LR - secure$.

What happens if we want to encrypt more than 1 bit? We'll find out next lecture!