U.C. Berkeley — CS276: Cryptography
Professors Luca Trevisan and David Wagner

Lecture 6: 2/8/02
Scribe: Naveen Sastry

Lecture 6: 2/8/02

Last Time

Last time we discussed four notions of security. We discussed attacks against malleability as well as message indistinguishibility. Furthermore, we looked at attackers with various power: those with the ability to encrypt the plain text of their choice (CPA) and those with access to decrypt the chosen cipher text of their choice (CCA). We developed the following relationships between these four classes:

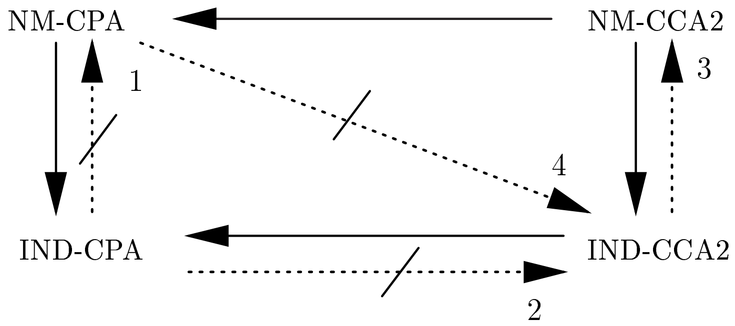

Figure 1: Last time, we discovered the solid relationships. Today, we will address (1) and (2); (3) and (4) are shown but not proved

Today, we will show that IND-CPA $\not\Rightarrow$ NM-CPA and IND-CPA $\not\Rightarrow$ IND-CCA2. Next time we will show the equivalence of IND-CCA2 and NM-CCA2 (i.e., IND-CCA2 $\Rightarrow$ NM-CCA2). Finally, we will state, but not prove, that NM-CPA $\not\Rightarrow$ IND-CCA2. Thus, we will separate the classes NM-CPA from IND-CPA and IND-CPA from IND-CCA2.

An Easy Separation

Theorem 1
IND-CPA $\not\Rightarrow$ NM-CPA

Proof: We will find an encryption scheme which obeys message independence but which is malleable. We note that we have not yet learned of a truly MI scheme since Goldwasser-Micali with RSA assumes that factoring is a hard trapdoor permutation, which has not yet been proved. So we'll change our statement slightly, so that if there exists any scheme which satisfies IND, we can show a separation.

Suppose (E, D, G) is a scheme which is IND-CPA. We will create a new scheme (E', D', G') which is IND-CPA but not NM-CPA.

Let $E'_{\mathrm{pk}}(x) = 0 \,||\, E_{\mathrm{pk}}(x)$, the old encryption prepended by the zero bit.

Let $D'_{\text{sk}}(b \mid\mid y) = D_{\text{sk}}(y)$, the old decryption which ignores the first bit.

We note that this is still message independent, since we have only prepended the encryption with a zero bit. Thus, (E', D', G') is still IND-CPA. However, it is not NM-CPA since an adversary can change the encrypted message to create a related message simply by changing the first bit to a 1. This messages is related under the equality relation since it decrypts to the same message $\square$

Another Separation

Theorem 2
IND-CPA $\not\Rightarrow$ IND-CCA2.

Proof: Consider the Goldwasser-Micali scheme under RSA:

$$E(x) = r^3 \pmod n \mid\mid s \mid\mid r \cdot s \oplus x$$

Suppose you have you have a decryption oracle $\mathcal{O}(y) = y^{1/3} \pmod n$ under CCA2. As always, you cannot ask the oracle about r^3 but can ask about:

$$\frac{1}{b}\mathcal{O}(b^3 \cdot y \pmod n)) = y^{1/3} \pmod n$$

Using Goldreich-Levin theorem, we pick s to be the unit vector so that we can decode the message.

Thus, the Goldwasser-Micali scheme is IND-CPA (from earlier, as long as RSA is a good trapdoor), but not IND-CCA2, showing a separation between these two classes. $\square$

Random Oracles

Random Oracles attempt to formalize intuitive heuristic design approaches. Hash functions approximate the behavior of a random oracle.

We define a random oracle $\mathcal{G} : \{0,1\}^k \to \{0,1\}^k$. We can alternatively define the output as a stream, so that the range is $\{0,1\}^\infty$. We can now outline an encryption scheme which has a reasonable chance of being secure (in fact, we will show that it is secure shortly):

$$E(x) = r^3 \pmod n \mid\mid \mathcal{G}(r) \oplus x$$

Thus, the encryption of a message produces a ciphertext which is an additive constant larger, and not a multiplicative constant as we have seen in prior schemes.

Definition 1 (Security) *An adversary is a pair (A_0, A_1) of algorithms with oracles $\mathcal{G}, \mathcal{O}$. The scheme is (t, q, ϵ) RO-IND-X secure if:*

$$\Pr_{\mathcal{G},(pk,\ sk)\in Gen}[A^{\mathcal{G},\mathcal{O}}(pk, m_0, m_1, E^{\mathcal{G}}(m_0)) = 0] - \Pr[A^{\mathcal{G},\mathcal{O}}(pk, m_0, m_1, E^{\mathcal{G}}(m_1) = 0)] \le \epsilon$$

where we define the oracle $\mathcal{O}$ as:

$$\mathcal{O} = \begin{cases} D^{\mathcal{G}}_{sk} & \text{if } X = CCA2 \\ - & \text{if } X = CPA \end{cases}$$

and $(m_0, m_1) \leftarrow A_0^{\mathcal{G},\mathcal{O}}(pk)$

THEOREM 3
$E(x) = T(r) \mid\mid \mathcal{G}(r) \oplus x$ is RO-IND-CPA for trapdoor T.

PROOF: The proof will be similar to our normal constructions. Suppose that the theorem is not true, and that $A = (A_0, A_1)$ are good adversaries with significant advantage ϵ. Suppose that $y = T(r)$. We will show how we can use this adversary to break the trapdoor, thus leading to a contradiction. A_0 is used to generate the plaintext of the messages (m_0, m_1). A_1 will then be handed the original plaintext of m_0 and m_1 as well as the encrypted version of the one of the two messages; it will be able to discern whether the encrypted message is m_0 or m_1.

Both A_0 and A_1 can make queries to the random oracle $\mathcal{G}$. Using algorithms A_0 and A_1, we can invert the trapdoor (without the key, of course) by utilizing the following construction. We provide A_0 and A_1 with a modified version of the random oracle $\mathcal{G}'$: namely, if either algorithm ever asks $\mathcal{G}$ regarding $r : T(r) = y$, then our random oracle will output r and we have inverted the trapdoor; on all other queries, our oracle behaves as any other random oracle would.

Furthermore, instead of feeding $E^{\mathcal{G}}(m_i) = T(r') \mid\mid \mathcal{G}(r') \oplus m_i$ to A_1, we send it $T(r) \mid\mid z$ where $z = \{0,1\}^{|m_i|}$, a random string value. Note that $r \neq r'$; rather, r is what we'd like to invert, while r' is the random value generated in the process of encrypting m_i. As before, we watch if A_1 queries for $r : T(r) = y$. We will show that A_1 *must* make such queries.

Let BAD be the event that A_0 and A_1 never query $\mathcal{G}'$ with $r = T^{-1}(y)$. By assumption, the probability that A succeeds is $1/2 + \epsilon$. Thus,

$$\begin{aligned} \Pr[\text{A succeeds}] &= \Pr[\text{A succeeds} \mid \text{BAD}]\Pr[\text{BAD}] + \Pr[\text{A succeeds} \mid \text{not BAD}]\Pr[\text{not BAD}] \\ 1/2 + \epsilon &< 1/2 + \Pr[\text{not BAD}] \\ \epsilon &< \Pr[\text{ not BAD}] \end{aligned}$$

since if the event BAD occurs, there is an equal likelihood of guessing the proper message (and it can do no better since all other queries to $\mathcal{G}'$ are uniform and independent of everything).

Therefore, we can invert the trapdoor T on an input of our choice with significant chance, thus leading to a contradiction. □

It is easy to see that the above scheme is not CCA2 since one can easily change a single bit of $G(r) \oplus x$ and ask the decryption oracle its value, and figure out a trivially related x. Thus, we can use another random oracle $\mathcal{H}$ to "protect" $\mathcal{G}(r) \oplus x$ from being modified.

THEOREM 4
$E(x) = T(r) \mid\mid \mathcal{G}(r) \oplus x \mid\mid \mathcal{H}(r \mid\mid x)$ is RO-IND-CCA2 for trapdoor T.

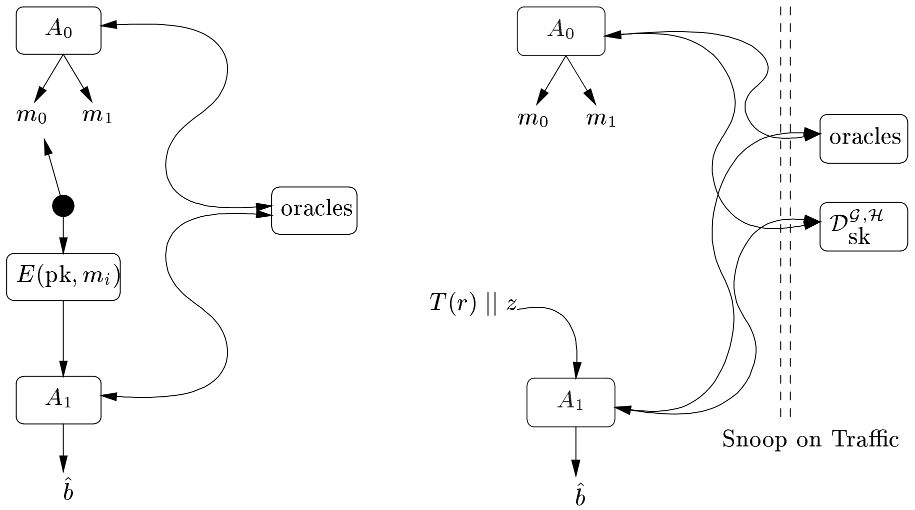

Figure 2: A sample construction which is used in the proofs for showing RO-IND-X. The left picture shows the "standard" adversary (A_0, A_1) as it attempts to distinguish between the encrypted forms of m_0 and m_1. In the right, we use A_0 and A_1 to invert $T(r)$ on r of our choice. We use special rules to simulate the oracles (by snooping on the traffic) as well as simulating the decryption oracle (for RO-IND-CCA2).

PROOF: We will utilize a similar construction as above. Assume that there exists an adversary (A_0, A_1) that succeeds with probability $1/2 + \epsilon$. We will utilize this algorithm to compute T^{-1} on a value of our choice.

In addition to receiving $\mathcal{G}$, both A_0 and A_1 will receive the random oracle $\mathcal{H}$ as well as the decryption oracle $\mathcal{D}^{\mathcal{G},\mathcal{H}}$. As before, we snoop the queries to $\mathcal{G}$ and $\mathcal{H}$, with $\mathcal{G}$ looking for queries to $r : f(r) = y$ and $\mathcal{H}$ looking for queries $r \mid\mid x : f(r) = y$. The decryption oracle is also modified so that queries for $a \mid\mid b \mid\mid c$ check for queries r to $\mathcal{G}$ or $r \mid\mid u$ to $\mathcal{H}$ such that $T(r) = y$ and $b = \mathcal{G}(r) \oplus u$, returing u if so. otherwise, it return invalid.

We then run A_0 as before; we then send $y = f(r) \mid\mid w \mid\mid z)$ for w, z random, to $A_1^{\mathcal{G},\mathcal{H},\mathcal{D}^{\mathcal{G},\mathcal{H}}}$, which we will trick into decrypting y.

As above, we'll define two events: Q [which is "not BAD2" from lecture] and LUCKY ["BAD1" from lecture]. Let LUCKY be the event A asks $\mathcal{D}^{\mathcal{G},\mathcal{H}}$ for a query $a \mid\mid b \mid\mid c$, with $c = \mathcal{H}(i)$, with $i = T^{-1}(a) \mid\mid b \oplus \mathcal{G}(T^{-1}(a))$, but never queries $\mathcal{H}$ for i. Letting n be the number of queries that the adversary makes against $\mathcal{D}^{\mathcal{G},\mathcal{H}}$, we can bound Pr[LUCKY] by $n2^{-|k|}$ since asking $\mathcal{D}^{\mathcal{G},\mathcal{H}}$ values which have not been inverted by $\mathcal{H}$ make no sense and can only be lucky guesses.

Let Q be the event that A_1 and A_0 asks $\mathcal{G}$ regarding r or $\mathcal{H}$ regarding rx. As above, we will show Pr[Q] is large.

In addition, Pr[A succeeds|not LUCKY and not Q] $= 1/2$ since the adversary has no information and can only guess. From above, we know that Pr[A succeeds] $> 1/2 + \epsilon$. Thus:

$$\Pr[\text{A succeeds}] \leq \Pr[\text{A succeeds}|\text{LUCKY}]\Pr[\text{LUCKY}]$$

$$\begin{aligned} & + \quad \Pr[\text{A succeeds} \mid \text{not } \textsc{lucky} \text{ and } \textsc{q}]\Pr[\text{not lucky and } \textsc{q}] \\ & + \quad \Pr[\text{A succeeds} \mid \text{not } \textsc{lucky} \text{ and not } \textsc{q}]\Pr[\text{not } \textsc{lucky} \text{ and not } \textsc{q}] \\ 1/2 + \epsilon & < \quad n2^{-k} + \Pr[\textsc{q}] + 1/2 \\ \epsilon - n2^{-k} & < \quad \Pr[\textsc{q}] \end{aligned}$$

Thus, there is a non-negligible fraction of the time in which A will reveal the inverse of our trapdoor value to either $\mathcal{G}$ or $\mathcal{H}$. □

References

[1] M. Bellare, P. Rogaway. *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols.* First ACM Conference on Computer and Communications Security. November, 1993.