CMSC 858K — Advanced Topics in Cryptography	February 12, 2004

Lecture 6

Lecturer: *Jonathan Katz* — Scribe(s): *Omer Horvitz, Zhongchao Yu, John Trafton, Akhil Gupta*

1 Introduction

In this lecture, we show how to construct a public-key encryption scheme secure against non-adaptive chosen-ciphertext attacks, given a semantically-secure public-key encryption scheme and an adaptively-secure non-interactive zero-knowledge proof system (in the common random string model).

2 Adaptively-Secure Non-Interactive Zero-Knowledge

We begin with a definition of a basic case of non-interactive zero-knowledge.

Definition 1 A pair of PPT algorithms $(\mathcal{P}, \mathcal{V})$ is a *non-interactive zero-knowledge* (*NIZK*) proof system for a language $L \in \mathrm{NP}$ if there exists some polynomial poly such that:

1. Completeness (for $x \in L$, $\mathcal{P}$ generates proofs that $\mathcal{V}$ accepts): For all $x \in L \cap \{0,1\}^k$ and all witnesses w for x,
$$\Pr[r \leftarrow \{0,1\}^{\mathrm{poly}(k)}; \Pi \leftarrow \mathcal{P}(r, x, w) : \mathcal{V}(r, x, \Pi) = 1] = 1.$$
2. Soundness (for $x \notin L$, no prover can generate proofs that $\mathcal{V}$ accepts with better than negligible probability): For all $x \in \{0,1\}^k \setminus L$ and all (possibly unbounded) algorithms $\mathcal{P}^*$, the following is negligible in k:
$$\Pr[r \leftarrow \{0,1\}^{\mathrm{poly}(k)}; \Pi \leftarrow \mathcal{P}^*(r, x) : \mathcal{V}(r, x, \Pi) = 1].$$
3. Zero-knowledge (for $x \in L$, the view of any verifier can be efficiently simulated without knowledge of a witness): There exists a PPT algorithm Sim such that for any $x \in L \cap \{0,1\}^k$ and any witness w for x, the following ensembles are computationally indistinguishable:
$$\begin{array}{ll} (1) & \left\{ r \leftarrow \{0,1\}^{\mathrm{poly}(k)}; \Pi \leftarrow \mathcal{P}(r, x, w) : (r, x, \Pi) \right\}_k \\ (2) & \left\{ (r, \Pi) \leftarrow \mathsf{Sim}(x) : (r, x, \Pi) \right\}_k . \end{array}$$

$\diamondsuit$

For our purposes, we need to strengthen the definition in two ways. In the soundness requirement, we would like to also protect against a prover who chooses $x \notin L$ *after* seeing the common random string r. In the zero-knowledge requirement, we would like to make the simulator's job a little harder by making it output a simulated common random string

r first, and only then supplying it with an $x \in L$ for which it needs to generate a simulated proof. In particular, such an x may be chosen adaptively (by an adversary, say) based on r. In the following, we use $\Pr_{\mathsf{E}}[\mathsf{A}]$ to denote the probability that event A occurs in the probabilistic experiment E.

Definition 2 A pair of PPT algorithms $(\mathcal{P}, \mathcal{V})$ is an *adaptive, non-interactive zero-knowledge* (*aNIZK*[1]) proof system for a language $L \in \mathrm{NP}$ if there exists a polynomial poly such that:

1. Completeness: Same as above.
2. Soundness (now, the cheating prover may choose $x \notin L$ *after* seeing r): For all (possibly unbounded) algorithms $\mathcal{P}^*$, the following is negligible in k:

$$\Pr\left[r \leftarrow \{0,1\}^{\mathrm{poly}(k)}; (x, \Pi) \leftarrow \mathcal{P}^*(r) : \mathcal{V}(r, x, \Pi) = 1 \wedge x \in \{0,1\}^k \setminus L\right].$$

3. Zero-knowledge (simulator must output r first, is then given x, and asked to produce a simulated proof): Let $(\mathsf{Sim}_1, \mathsf{Sim}_2)$, (A_1, A_2) be a pair of two-staged algorithms (we may assume that the first stage outputs some state information which is passed as input to the second stage; this will be implicit). Consider the following experiments:

Game $\mathsf{ZK}_{\mathrm{real}}$	Game $\mathsf{ZK}_{\mathrm{sim}}$
$r \leftarrow \{0,1\}^{\mathrm{poly}(k)}$	$r \leftarrow \mathsf{Sim}_1(1^k)$
$(x, w) \leftarrow A_1(r) \quad (x \in L \cap \{0,1\}^k)$	$(x, w) \leftarrow A_1(r) \quad (x \in L \cap \{0,1\}^k)$
$\Pi \leftarrow \mathcal{P}(r, x, w)$	$\Pi \leftarrow \mathsf{Sim}_2(x)$
$b \leftarrow A_2(r, x, \Pi)$	$b \leftarrow A_2(r, x, \Pi)$

We require that there exist a PPT *simulator* $(\mathsf{Sim}_1, \mathsf{Sim}_2)$ such that for any PPT algorithm (A_1, A_2) the following is negligible in k:

$$\left|\Pr_{\mathsf{ZK}_{\mathrm{real}}}[A_2 \text{ outputs } 0] - \Pr_{\mathsf{ZK}_{\mathrm{sim}}}[A_2 \text{ outputs } 0]\right|.$$

Equivalently, the "real" game $\mathsf{ZK}_{\mathrm{real}}$ is computationally indistinguishable from the "simulated" game $\mathsf{ZK}_{\mathrm{sim}}$. (In other words, the adversary cannot tell whether it is participating in the first or the second experiment (except with negligible probability). A_2's output can be thought of as its guess towards which experiment it is in. This means that the simulator is able to simulate the adversary's view in the real execution.)

$\diamondsuit$

To simplify the notation a little, we will usually drop the stage identifier for A; when we refer to A's output in the experiment, we will mean A_2's output.

3 A Public-Key Encryption Scheme Secure Against Non-Adaptive Chosen-Ciphertext Attacks

Let $(\mathsf{Gen}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme and $(\mathcal{P}, \mathcal{V})$ be an adaptively-secure NIZK proof system for languages in NP. The following construction is due to Naor and Yung [1].

[1] This notation is non-standard.

$\underline{\mathsf{Gen}^*(1^k)}$	$\underline{\mathcal{E}^*_{(pk_1,pk_2,r)}(m)}$	$\underline{\mathcal{D}^*_{sk_1}(c_1,c_2,\Pi)}$

$(pk_1,sk_1) \leftarrow \mathsf{Gen}(1^k)$;
$(pk_2,sk_2) \leftarrow \mathsf{Gen}(1^k)$;
$r \leftarrow \{0,1\}^{\text{poly}(k)}$;
$pk^* = (pk_1,pk_2,r)$;
$sk^* = sk_1$
	$w_1,w_2 \leftarrow \{0,1\}^*$;
	$c_1 = \mathcal{E}_{pk_1}(m;w_1)$;
	$c_2 = \mathcal{E}_{pk_2}(m;w_2)$;
	$\Pi \leftarrow \mathcal{P}(r,(c_1,c_2),(w_1,w_2,m))$;
	Output (c_1,c_2,Π)
		If $\mathcal{V}(r,(c_1,c_2),\Pi) = 0$
		   Output $\perp$;
		else
		   Output $\mathcal{D}_{sk_1}(c_1)$

A few words of explanation are due here. For key generation, we use the underlying key-generation algorithm to produce two pairs of (public, private) keys, publish the public keys and a random string r (to serve as the common random string for the proof system), and keep the first underlying private key as our private key (we discard the second private key). For encryption, we use the underlying algorithm to encrypt the given message m *twice*, under both pk_1 and pk_2, with the random tapes of the encryption algorithm fixed to w_1, w_2, respectively.[2] (For a probabilistic algorithm $A(\cdot)$, the notation $A(\cdot;w)$ is used to denote that A's random tape is fixed to a particular $w \in \{0,1\}^*$.) We then use our prover to generate a proof that (c_1,c_2) are encryptions of the same message under pk_1, pk_2 in the underlying scheme; i.e., $(c_1,c_2) \in L$ where

$$L = \{(c_1,c_2)\,|\exists m,w_1,w_2 \text{ such that } c_1 = \mathcal{E}_{pk_1}(m;w_1), c_2 = \mathcal{E}_{pk_2}(m;w_2)\}\,,$$

using w_1, w_2, and m as witnesses. Note that $L \in$ NP. We send the ciphertexts and the proof to the receiver. For decryption, we use the underlying decryption algorithm on the first ciphertext, if the provided proof verifies correctly.

Theorem 1 *Assuming that* $(\mathsf{Gen},\mathcal{E},\mathcal{D})$ *is semantically secure and that* $(\mathcal{P},\mathcal{V})$ *is an adaptively-secure NIZK proof system,* $(\mathsf{Gen}^*,\mathcal{E}^*,\mathcal{D}^*)$ *is secure against non-adaptive ("lunchtime"[3]) chosen-ciphertext attacks (i.e., is CCA1 secure).*

The remainder of these notes is the beginning of a proof of this theorem (we continue the proof next lecture). Let A be a two-staged PPT algorithm, where the first stage has access to an oracle. Consider the following two experiments (their differences are in boldface):

Game $\mathsf{CCA1}_0$	Game $\mathsf{CCA1}_1$
$(pk_1,sk_1),(pk_2,sk_2) \leftarrow \mathsf{Gen}(1^k)$ $r \leftarrow \{0,1\}^{\text{poly}(k)}$ $pk^* = (pk_1,pk_2,r)$, $sk^* = sk_1$ $(m_0,m_1) \leftarrow A^{\mathcal{D}^*_{sk^*}(\cdot)}(pk^*)$ $w_1,w_2 \leftarrow \{0,1\}^*$ $c_1 = \mathcal{E}_{pk_1}(\boldsymbol{m_0};w_1)$, $c_2 = \mathcal{E}_{pk_2}(\boldsymbol{m_0};w_2)$ $\Pi \leftarrow \mathcal{P}(r,(c_1,c_2),(w_1,w_2,\boldsymbol{m_0}))$ $b \leftarrow A(pk^*,c_1,c_2,\Pi)$	$(pk_1,sk_1),(pk_2,sk_2) \leftarrow \mathsf{Gen}(1^k)$ $r \leftarrow \{0,1\}^{\text{poly}(k)}$ $pk^* = (pk_1,pk_2,r)$, $sk^* = sk_1$ $(m_0,m_1) \leftarrow A^{\mathcal{D}^*_{sk^*}(\cdot)}(pk^*)$ $w_1,w_2 \leftarrow \{0,1\}^*$ $c_1 = \mathcal{E}_{pk_1}(\boldsymbol{m_1};w_1)$, $c_2 = \mathcal{E}_{pk_2}(\boldsymbol{m_1};w_2)$ $\Pi \leftarrow \mathcal{P}(r,(c_1,c_2),(w_1,w_2,\boldsymbol{m_1}))$ $b \leftarrow A(pk^*,c_1,c_2,\Pi)$

To prove the scheme CCA1 secure, we need to show that A cannot distinguish the above two games; i.e.,to show that the following is negligible: $|\Pr_{\mathsf{CCA1}_0}[b=0] - \Pr_{\mathsf{CCA1}_1}[b=0]|$.

[2]The notation $w_1 \leftarrow \{0,1\}^*$ just means that a "long enough" random string is chosen.

[3]That is, the adversary is assumed to be able to "play" with the decryption oracle while people are out for lunch, but not afterward when he gets the challenge ciphertext.

To that effect, we introduce a sequence of intermediate games, and show that A cannot distinguish each game from its subsequent one; the theorem will then follow.

Let $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ be the simulator for our proof system. In the first game, we replace the random string and legitimate proof of game $\mathsf{CCA1}_0$ with a simulated random string and simulated proof. Once again, the differences between the new game and game $\mathsf{CCA1}_0$ are highlighted.

Game 1
$(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{Gen}(1^k)$
$r \leftarrow \boldsymbol{\mathsf{Sim}_1(1^k)}$
$pk^* = (pk_1, pk_2, r),\ sk^* = sk_1$
$(m_0, m_1) \leftarrow A^{\mathcal{D}^*_{sk^*}(\cdot)}(pk^*)$
$w_1, w_2 \leftarrow \{0,1\}^*$
$c_1 = \mathcal{E}_{pk_1}(m_0; w_1),\ c_2 = \mathcal{E}_{pk_2}(m_0; w_2)$
$\Pi \leftarrow \boldsymbol{\mathsf{Sim}_2((c_1, c_2))}$
$b \leftarrow A(pk^*, c_1, c_2, \Pi)$

Claim 2 $|\Pr_{\mathsf{CCA1}_0}[A \text{ outputs } 0] - \Pr_1[A \text{ outputs } 0]|$ *is negligible.*

Proof By reduction to the zero-knowledge property of the proof system: we use A to construct an algorithm B that attempts to distinguish real from simulated proofs.

$B(1^k)$
Receive r as first-stage input;
$(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{Gen}(1^k)$;
$pk^* = (pk_1, pk_2, r)$;
$sk^* = sk_1$;
$(m_0, m_1) \leftarrow A^{\mathcal{D}^*_{sk^*}(\cdot)}(pk^*)$; // note that B has no trouble
// simulating the decryption oracle for A
$w_1, w_2 \leftarrow \{0,1\}^*$;
$c_1 = \mathcal{E}_{pk_1}(m_0; w_1)$, $c_2 = \mathcal{E}_{pk_2}(m_0; w_2)$;
Output $((c_1, c_2), (w_1, w_2, m_0))$ as first-stage output;
Receive Π as second-stage input;
$b \leftarrow A(pk^*, c_1, c_2, \Pi)$;
Output b as second-stage output.

Now, when the inputs to B are a random string r and a real proof Π, then A's view in the above experiment is precisely its view in game $\mathsf{CCA1}_0$, and so $\Pr_{\mathsf{ZK}_{\text{real}}}[B \text{ outputs } 0] = \Pr_{\mathsf{CCA1}_0}[A \text{ outputs } 0]$. On the other hand, when the inputs to B are a simulated string and a simulated proof, A's view in B is precisely its view in game 1, and so $\Pr_{\mathsf{ZK}_{\text{sim}}}[B \text{ outputs } 0] = \Pr_1[A \text{ outputs } 0]$. Since $|\Pr_{\mathsf{ZK}_{\text{real}}}[B \text{ outputs } 0] - \Pr_{\mathsf{ZK}_{\text{sim}}}[B \text{ outputs } 0]|$ is negligible (since the proof system is adaptively-secure NIZK), we have that

$$|\Pr_{\mathsf{CCA1}_0}[A \text{ outputs } 0] - \Pr_1[A \text{ outputs } 0]|$$

is negligible as well. ■

The second game differs from game 1 in that it does not double-encrypt m_0, but instead encrypts m_0 once and m_1 once.

Game 2
$(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{Gen}(1^k)$
$r \leftarrow \mathsf{Sim}_1(1^k)$
$pk^* = (pk_1, pk_2, r),\ sk^* = sk_1$
$(m_0, m_1) \leftarrow A^{\mathcal{D}^*_{sk^*}(\cdot)}(pk^*)$
$w_1, w_2 \leftarrow \{0,1\}^*$
$c_1 = \mathcal{E}_{pk_1}(m_0; w_1),\ \boldsymbol{c_2 = \mathcal{E}_{pk_2}(m_1; w_2)}$
$\Pi \leftarrow \mathsf{Sim}_2((c_1, c_2))$
$b \leftarrow A(pk^*, c_1, c_2, \Pi)$

Note that in the above, the simulator is given as input encryptions of two *different* messages. Such an input is *not* in L, and in general there is not much we can say about the simulator's output in this case. However, we will see that in this particular case the game is indistinguishable to A because the semantic security of the underlying encryption scheme implies that encryptions of m_0 are indistinguishable from encryptions of m_1. Of course, this will require a formal proof.

Claim 3 $|\Pr_1[A \text{ outputs } 0] - \Pr_2[A \text{ outputs } 0]|$ *is negligible.*

Proof We use A to construct B that attempts to break the semantic security of the underlying scheme. Recall that B is given a public key, outputs two messages (m_0, m_1), is given the encryption of one of these, and has to guess which one. But B does *not* have access to a decryption oracle.

$B(pk)$
Set $pk_2 = pk$;
$(pk_1, sk_1) \leftarrow \mathsf{Gen}(1^k)$;
$r \leftarrow \mathsf{Sim}_1(1^k)$;
$pk^* = (pk_1, pk_2, r),\ sk^* = sk_1$;
$(m_0, m_1) \leftarrow A^{\mathcal{D}^*_{sk^*}(\cdot)}(pk^*)$; // note that B has no trouble
// simulating the decryption oracle for A
Output (m_0, m_1);
Receive c_2 (an encryption of either m_0 or m_1 using (unknown) random tape w_2);
$w_1 \leftarrow \{0,1\}^*$;
$c_1 = \mathcal{E}_{pk_1}(m_0; w_1)$;
$\Pi \leftarrow \mathsf{Sim}_2((c_1, c_2))$;
$b \leftarrow A(pk^*, c_1, c_2, \Pi)$;
Output b.

Now, when c_2 is an encryption of m_0, then A's view above is precisely its view in game 1. On the other hand, when c_2 is an encryption of m_1, then A's view above is precisely its view in game 2. Therefore, the probability that A distinguishes game 1 from game 2 is precisely the probability that B distinguishes an encryption of m_0 from an encryption of m_1, which is negligible by the semantic security of the underlying encryption scheme. ∎

In the same way as above, we would now like to "switch" c_1 from being an encryption of m_0 to being an encryption of m_1. Here, however, a potential problem arises! To prove a claim analogous to Claim 3, we would need to construct some adversary B that gets $pk = pk_1$ and then has to distinguish whether the ciphertext c_1 it receives is an encryption of m_0 or m_1. But, in order to do this it has to somehow simulate a decryption oracle for A — and this seems to require sk_1, which B does not have! (If B has sk_1 then it would be easy for B to break semantic security of the scheme....) So, we will have to do a little more work before continuing.

Let Fake be the event that A submits a query (c_1, c_2, Π) to its decryption oracle (in stage 1) such that $\mathcal{D}_{sk_1}(c_1) \neq \mathcal{D}_{sk_2}(c_2)$ but $\mathcal{V}(r, (c_1, c_2), \Pi) = 1$. Note that Π is then a valid-looking proof for a false statement (since $(c_1, c_2) \notin L$).

Claim 4 $\Pr_2[\mathsf{Fake}]$ *is negligible.*

Proof First, note that $\Pr_2[\mathsf{Fake}] = \Pr_1[\mathsf{Fake}]$. This is because A submits oracle queries only in its first stage, and up to that stage the games are *identical*.

Next, we show that $|\Pr_1[\mathsf{Fake}] - \Pr_{\mathsf{CCA1}_0}[\mathsf{Fake}]|$ is negligible. Up to A's first stage, the games differ only in r being a random string or a simulated string. Construct an algorithm B that attempts to distinguish random from simulated strings, as follows:

$B(r)$
  $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{Gen}(1^k)$;
  $pk^* = (pk_1, pk_2, r)$;
  Run $A^{\mathcal{D}^*_{sk^*}(\cdot)}(pk^*)$, simulating the oracle for A normally except that
    if for any decryption query (c_1, c_2, Π) it is the case that
    $\mathcal{V}(r, (c_1, c_2), \Pi) = 1$ but $\mathcal{D}_{sk_1}(c_1) \neq \mathcal{D}_{sk_2}(c_2)$,
    then output 1 and stop (note that now B does *not* throw away sk_2);
  Otherwise, once A is done with its first stage simply output 0

Now, $\Pr_{\mathsf{ZK}_{\mathrm{sim}}}[B \text{ outputs } 0] = \Pr_1[\mathsf{Fake}]$. Similarly, $\Pr_{\mathsf{ZK}_{\mathrm{real}}}[B \text{ outputs } 0] = \Pr_{\mathsf{CCA1}_0}[\mathsf{Fake}]$. Since $|\Pr_{\mathsf{ZK}_{\mathrm{real}}}[B \text{ outputs } 0] - \Pr_{\mathsf{ZK}_{\mathrm{sim}}}[B \text{ outputs } 0]|$ is negligible (since the proof system is adaptively-secure NIZK), we have that $|\Pr_1[\mathsf{Fake}] - \Pr_{\mathsf{CCA1}_0}[\mathsf{Fake}]|$ is negligible as well.

Finally, note that $\Pr_{\mathsf{CCA1}_0}[\mathsf{Fake}]$ is negligible. This is because Fake occurs when A produces a valid proof for a $(c_1, c_2) \notin L$, which can only happen with a negligible probability because of the soundness of the NIZK proof system (note that r now is a truly random string). The claim follows. ■

We pick up the proof from here in the next lecture. Informally, the next game we introduce differs from game 2 only in that sk_2 is used for decryption instead of sk_1. The adversary's view in the games only differs if Fake occurs, which happens with negligible probability. All that's left to be done is to switch c_1 to an encryption of m_1 (similar to the introduction of game 2), switch the decryption key back to sk_1, and then go back to a random string and a real proof (similar to the introduction of game 1). This gets us back to game $\mathsf{CCA1}_1$ as desired, and will complete the proof.

References

[1] M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks. In *Proceedings of the ACM Symposium on the Theory of Computing*, pages 427-437, 1990.