CSE 599b: Cryptography **(Winter 2006)**

Lecture 10: Secure Symmetric Encryption

3 February 2006

Lecturer: Paul Beame *Scribe: Paul Beame*

1 Parallel Computation of Pseudorandom Function Families

All the constructions we have shown so far are quite sequential. In order to derive pseudorandom generators of output length m we composed one-way permutations m times; to build PRFFs on length n strings from pseudorandom generators we composed generators n times. The total amount of work is an issue but more important is the delay before values are produced. (Recall that DES was particularly fast in hardware because it could be computed in small circuit depth, which is equivalent to fast parallel time.)

In a series of papers Naor and Reingold showed a general method to achieve PRFFs and PRPFs that can be computed in $O(\log^2 n)$ parallel time ($O(logn)$ depth of integer multiplication.) or even $O(\log n)$ parallel time (complexity classes NC^1 or TC^0). (Some of the constructions involve assumptions similar to those involved in Blum Squaring, which is a candidate one-way function that requires only one modular multiplication.)

One of the primitives they introduce is the *pseudo-random synthesizer* which is a compression function that has properties that make it a particularly useful form of pseudorandom generator: $S : \{0,1\}^{2k} \to \{0,1\}^k$ is a synthesizer if and only if for two sequences $x_1, \ldots, x_k$, $y_1, \ldots, y_k \leftarrow \mathcal{U}_k$, the string of all $S(x_i, y_j)$ is indistinguishable from a string from $\mathcal{U}_{k^3}$. These synthesizers are directly constructed based on hypothesized hard functions.

This leads to a parallel construction of PRFFs as follows: (Assume for simplicity that k is a power of 2.) The key K is viewed as a length $2k^2$ string $a_1^0, a_1^1, \ldots, a_k^0, a_k^1$. The value of F_K on input $x \in \{0,1\}^k$ is found by creating a binary tree circuit of height $\log_2 k$ with the i-th leaf labeled by $a_i^{x_i}$ and the value of of each node being the synthesizer S applied to its children.

Very recently, Applebaum, Ishai, and Kushilevitz have shown that one can do even better for certain primitives. Namely they show that if PRNGs exist of the type that Naor and Reingold propose then they can be converted to a PRNG from n bits to $n + n^\epsilon$ bits that operate in NC^0, the class of functions in which each output bit depends on only a *constant* number of input bits! This is a very surprising result. The S-boxes in DES are natural example of this kind of computation but it had been conjectured previously that there could be no provable security based on such weak forms of computation.

(Note: These papers by Naor and Reingold and Applebaum, Ishai, and Kushilevitz would be good candidates for presentations.)

2 Security Definitions for Cryptographic Systems

When we talk about security of cryptosystems there are two somewhat orthogonal aspects that must be considered: The types of cryptographic attacks allowed and the kind of security against these attacks.

2.1 Attacks

We describe the kind attacks allowed in the context of encryption but these general classifications can be applied more broadly. For some of the following there is a goal of finding information about a particular ciphertext which we call the 'challenge ciphertext'.

1. Ciphertext-only (Passive Attack) The adversary has access to the ciphertext of some messages chosen from a known distribution.

2. Known Plaintext Attack (Passive Attack) The adversary has access to some $(plaintext, cipertext)$ pairs not of its choosing but from a known distribution.

    - e.g., this attack was used as part of the British effort in breaking the Enigma machine codes used by Germany during WWII. It was known that each day's message began with a weather report and the weather that actually occurred became known after the fact.

3. Chosen Plaintext Attack (CPA) The adversary sees $(plaintext, ciphertext)$ pairs where the plaintexts are of its choosing. The most general form of this attack is adaptive, in which the choice of each successive plaintext chosen may depend on the encryptions of the previous plaintexts.

    - The general idea is that the adversary can interact with the sender through other channels to cause the sender to send messages that will be useful to the adversary in decrypting other messages. E.g., during WWII, just prior to the battle of Midway the U.S. used this kind of attack to learn of the Japanese plans to attack the island. The U.S. had partially broken a Japanese code and suspected that a frequently mentioned place name was Midway. They also knew that the Japanese had broken a U.S. code but were not aware that the U.S. knew that the code was insecure. The U.S. sent a message using the broken code that there was a water shortage on Midway and were able to observe that the same reference was used as the Japanese reported the water shortage, confirm that Midway was the location in imminent danger, and reinforce the island.

4. Chosen Ciphertext Attack (CCA) The adversary sees $(plaintext, ciphertext)$ pairs where the ciphertexts are of its choosing (but distinct from the challenge ciphertext). There are two different forms of this attack:

    (a) Known as CCA-Pre, CCA1, or Lunchtime Attacks: The adversary's choices of ciphertext are not allowed to be a function of the challenge ciphertext and thus can be thought of as being made before receipt of the challenge ciphertext.

(b) Known as CCA-Post or CCA2: The adversary's choices of ciphertexts may depend on the challenge ciphertext.

- A simple form of these attacks occurs when the adversary sends additional messages to the receiver and observes the receiver's reactions. These can be modified forms of ciphertexts it has seen, or simply fake ciphertexts. A more specific example occurs in public-key identification in which parties prove their identity by decrypting challenge texts known as *nonces*. Another situation where this might occur (motivating the name Lunchtime Attacks) is if the adversary party can obtain access to decrypting machinery for a short period of time while everyone else is at lunch.

These are not the only forms of attacks possible in cryptographic settings. There include:

- Hardware attacks, in which the encryption and decryption devices themselves or the connection between them and the sender or receiver are attacked.
- Timing attacks, in which the time it takes to do cryptographic operations is analyzed.
- Traffic analysis, in which the pattern and volume of traffic between parties is analyzed.
- Network attacks such as Man-in-the-middle attacks, in which the adversary impersonates other parties and modifies the communications sent between parties.
- Exploiting insecure programming practices such as buffer over-runs, etc.

We cannot say anything useful about some of these attacks. Timing attacks can be easily avoided by slowing each operation so that it independent of its input (although this may slow computation somewhat). For us the most serious will be network attacks such as man-in-the-middle attacks. These are important but we will model many of the ill effects of these attacks using the main notions of attack defined above such as the CCA1 and CCA2 attacks.

2.2 Notions of security

1. Semantic Security (SS, SEM) [Goldwasser, Micali] Given any reasonable distribution (polynomial-time samplable distribution) on plaintexts and any polynomial-time computable property of plaintexts, the difference in probability of a PPT predicting the property of the plaintext M before and after seeing its ciphertext is negligible.

   Formally, to have a single algorithm used in the situations of 'before' and 'after' the algorithm takes as input a ciphertext-length string which is given by an encryption of a random plaintext M' in the 'before' case and an encryption of the plaintext M in the 'after' case.

2. Indistinguishability Security (IND) [Goldwasser, Micali] For any two messages M^0, M^1 of the adversary's choice, the adversary cannot distinguish between encryptions of M^0 and encryptions of M^1 is non-negligible. (In the case of CPA attacks, the choices of M^0 and M^1 can depend on everything that the adversary has seen.) Goldwasser and Micali showed that this much simpler definition is equivalent to Semantic Security.

3. Non-malleability (NM) Given ciphertext corresponding to plaintext M, the adversary cannot output the ciphertext of any plaintext M' for which the adversary knows the relationship between M and M' (but not necessarily either M or M').

We describe the kind of security using a notation that is a combination of the kind of security with the kind of attacks under which it holds. We thus get combinations such as IND-CPA, SS-CPA, or SEM-CPA, IND-CCA1, SS-CCA-Post, NM-CPA, etc.

Non-malleability is a more general property than semantic security. One way to compare them is that in semantic security, given $C = E(M)$ it is hard to find an M' that is related to M. In non-malleability, given $C = E(M)$ it is hard to find an $E(M')$ where M' is related to M. In fact, the first public-key cryptosystem shown to be IND-CPA (=SS-CPA) secure (also due to Goldwasser and Micali) is highly malleable. It has the property that each bit of the message was encrypted as a separate block of bits and these bits could be recombined and re-used.

3 Secure Symmetric Encryption

Definition 3.1. *A* symmetric encryption scheme *is a triple $(\mathcal{K}, \mathcal{E}, D)$ of PPT algorithms (D can be taken to be deterministic) such that*

- *$\mathcal{K}(1^k)$ produces a key $k \in \{0,1\}^k$; i.e. the keys are chosen according to a polynomial-time samplable distribution.*
- *For each k, the encryption function $\mathcal{E}$ takes a key K from $\{0,1\}^k$ output by $\mathcal{K}(1^k)$ and maps $M \in \{0,1\}^{\ell(k)}$ to it encryption $\mathcal{E}_K(M) = C \in \{0,1\}^{\ell'(k)}$. The encryption $\mathcal{E}_K(M)$ may not be unique given K. Moreover the encryption may be stateful, i.e. dependent on values from the previous times it has been called.*
- *The decryption function D has the property that for any K output by $\mathcal{K}$ and $C = \mathcal{E}_K(M)$, $D(K, C) = D_K(C) = M$.*

We now define IND-CPA security of a symmetric encryption scheme formally. For convenience we will consider PPT algorithms that call a subroutine (oracle) on a pair of inputs; we call these pair oracle PPTs.

Definition 3.2. *A symmetric encryption scheme $(\mathcal{K}, \mathcal{E}, D)$ is* IND-CPA secure *if and only if for every (pair oracle) PPT A, the function*

$$\epsilon(k) = \Pr[A^{\mathcal{E}_K(Select(\cdot,\cdot,0))}(1^k) = 1 \mid K \leftarrow \mathcal{K}(1^k)] - \Pr[A^{\mathcal{E}_K(Select(\cdot,\cdot,1))}(1^k) = 1 \mid K \leftarrow \mathcal{K}(1^k)],$$

is negligible where $Select(M^0, M^1, b) = M^b$ for $b \in \{0,1\}$.

That is, A gets an oracle that produces an encryption of one of its two inputs but it does not know which one. This formulation clearly allows A to receive encryptions of chosen plaintext M by calling its oracle with the pair (M, M) and later call it on the pair (M^0, M^1) that it wants to distinguish.

Just as in the case of the alternative versions of predictability in the hard-core bits definition we obtain the following alternative property for all pair oracle PPT A that implies IND-CPA security:

$$\Pr[A^{\mathcal{E}_K(Select(\cdot,\cdot,b))}(1^k) = b \mid K \leftarrow \mathcal{K}(1^k); b \leftarrow \mathcal{U}_1] \leq 1/2 + \epsilon(k)/2$$

where ϵ is negligible.

We will use this definition to analyze some of the ways for converting our ideal block ciphers: PRFFs and PRPFs to symmetric encryptions schemes.

3.1 Electronic Code Book (ECB)

With this method given a PRPF $\mathcal{F}$, $\mathcal{E}_K(M) = F_K(M)$. This obviously fails IND-CPA security by an algorithm A with $\epsilon(k) = 1$ that calls its oracle on (M, M) and then (M, M') for $M' \neq M$, outputting 1 if the answer is different in the two cases.

3.2 XOR Cipher

Let $\mathcal{F}$ be a PRFF with length $\ell(k)$. Define the symmetric encryption scheme $CTR_{\mathcal{F}}$ to be a scheme in which the $\mathcal{K}$ is defined based as for the PRFF, the i-th message $M_i \in \{0,1\}^{\ell(k)}$ is encrypted by $C_i = M_i \oplus F_K(i)$ where i is represented by its k-bit encoding, and decrypted by computing $C_i \oplus F_K(i)$.

(This is very much like using the PRFF as a PRNG and using he result as a stream cipher in place of a one-time pad.)

Theorem 3.3. *If $\mathcal{F}$ is a PRFF of length $\ell(k) = k$ then $CTR_{\mathcal{F}}$ is IND-CPA secure.*

Proof. Let A by any pair oracle PPT that using the alternative definition of IND-CPA security for $CTR_{\mathcal{F}}$ predicts the bit b with probability $= 1/2 + \epsilon(k)/2$.

We define an oracle PPT B that distinguishes $\mathcal{F}$ from a random element of $\mathcal{F}unc(k,k)$ as follows:

On input 1^k and oracle for f, B does the following:

1. Choose $b \leftarrow \mathcal{U}_1$.
2. Simulate A on input 1^k and whenever A makes its i-th call (M_i^0, M_i^1) to its oracle, calls the oracle f on input i and return $C_i = M_i^b \oplus f(i)$.
3. Output 1 if and only if A outputs b' with $b' = b$.

Observe that by construction

$$\Pr[B^{\mathcal{F}_k}(1^k) = 1] = \Pr[A^{\mathcal{E}_K(Select(\cdot,\cdot,b))}(1^k) = b \mid K \leftarrow \mathcal{K}(1^k); b \leftarrow \mathcal{U}_1] \geq 1/2 + \epsilon(k)/2.$$

Furthermore if B's oracle is chosen from $\mathcal{F}unc(k,k)$ then each $f(i)$ is chosen from the uniform distribution $\mathcal{U}_k$ so C_i is also distributed as $\mathcal{U}_k$, independent of whether $b = 0$ or $b = 1$. Therefore

the probability in step 3 of B, that A predicts b is precisely 1/2 so the advantage of B,

$$\begin{aligned}\epsilon'(k) &= \Pr[B^{\mathcal{F}_k}(1^k) = 1] - \Pr[B^{\mathcal{F}unc(k,k)}(1^k) = 1] \\ &= 1/2 + \epsilon(k)/2 - 1/2 \\ &= \epsilon(k)/2.\end{aligned}$$

Thus since $\epsilon'(k)$ must be negligible by the PRFF property of $\mathcal{F}$, $\epsilon(k)$ must be negligible as required for the IND-CPA security of $CTR_{\mathcal{F}}$. □

3.3 Random XOR Cipher

Typically, in using symmetric encryption, a message consisting of a series of blocks are encrypted at once and then later another message is sent, etc. In this context it is a little inconvenient for the parties to remember the state in $CTR_{\mathcal{F}}$ between messages.

A randomized version of the above XOR scheme resolves this problem. Instead of always starting at index 0, at the beginning of each message the sender chooses a random integer $R \in \{0,1\}^k$, and uses it as the first value of the counter, sending $\mathcal{E}_K(M_1 \dots M_m) = (R, C_1, \dots, C_m)$ where $C_i = M_i \oplus F_K(R + i)$.

This is easily seen to be IND-CPA secure using similar ideas to the above proof except that in addition to the probability of $\epsilon(k)/2$ of breaking the PRFF there is the chance that the regions of where the function f is queried overlap. If there are a total of $q(k)$ blocks sent for some polynomial q, then the chance that each of the at most $q(k)$ choices of R creates an overlap is at most $q(k)/2^k$. In the worst case B does not distinguish the possibilities for f in case of an overlap so the total probability of breaking the PRFF is at least $\epsilon(k)/2 - q(k)^2/2^k$ which is non-negligible if and only if $\epsilon(k)$ is.

3.4 Cipher Block Chaining with Random IV

For a PRPF $\mathcal{F}$ of length ℓ and an initialization vector IV, define the symmetric encryption scheme $CBC^{\mathcal{U}}_{\mathcal{F}}$ as follows: $\mathcal{K}$ is the key generation algorithm for $\mathcal{F}$; $\mathcal{E}_K(M_1 \dots M_m)$ is given by $C_0 C_1 \dots C_m$ where $C_0 = IV \leftarrow \mathcal{U}_\ell(k)$, and $C_i = F_K(C_{i-1} \oplus M_i)$. $D_K(C_0 C_1 \dots C_m) = F_K^{-1}(C_i) \oplus C_{i-1}$ for $i \geq 1$.

Theorem 3.4. *If $\mathcal{F}$ is a PRPF with length $\ell(k)$ at least $k^{\Omega(1)}$, $CBC^{\mathcal{U}}_{\mathcal{F}}$ is IND-CPA secure.*

Proof. Let A by any pair oracle PPT that using the alternative definition of IND-CPA security for $CBC^{\mathcal{U}}_{\mathcal{F}}$ predicts the bit b with probability $\geq 1/2 + \epsilon(k)/2$.

As in our previous proof we define an oracle PPT B that on input 1^k does the following:

1. Choose $b \leftarrow \mathcal{U}_1$.
2. Simulate A on input 1^k and whenever A makes a call $(M^0, M^1) = (M^0_1 M^0_2 \dots M^0_m, M^1_1 M^1_2 \dots M^1_m)$ to its oracle:
   (a) Choose $C_0 = IV \leftarrow \mathcal{U}_{\ell(k)}$

(b) For $i = 1$ to m compute $C_i = f(C_{i-1} \oplus M_i^b)$ where f is B's oracle.

(c) Return $C_0 C_1 \ldots C_m$ which we write as $CBC_f^{IV}(M^b)$ for convenience.

3. Output 1 if and only if A outputs b' with $b' = b$.

By construction, as in the previous proof,

$$\Pr[B^{\mathcal{F}_k}(1^k) = 1] = \Pr[A^{\mathcal{E}_K(Select(\cdot,\cdot,b))}(1^k) = b \mid K \leftarrow \mathcal{K}(1^k); b \leftarrow \mathcal{U}_1] = 1/2 + \epsilon(k)/2.$$

Furthermore, if B's oracle f is chosen from $\mathcal{F}unc(\ell(k), \ell(k))$ then provided that over all the $q(k)$ blocks that appear in calls to the oracle f none of the potential parameters $C_{i-1} \oplus M_i^0$ nor the $C_{i-1} \oplus M_i^1$ repeats an earlier call then each C_i will be uniformly random from $\mathcal{U}_{\ell(k)}$. Given distinct arguments the output C_i are all independently chosen from $\mathcal{U}_{\ell(k)}$ and thus the probability that there exists a collision involving a block of an M^0 or an M^1 is at most $2\binom{q(k)}{2}/2^{\ell(k)} \leq q(k)^2/2^{\ell(k)}$. Conditioned on there being are no collisions, A receives exactly the same distribution in its oracle on both $b = 0$ and $b = 1$ so it has probability precisely $1/2$ of guessing b in this case. Therefore

$$\Pr[B^{\mathcal{F}unc(\ell(k),\ell(k))}(1^k) = 1] \leq 1/2 + q(k)^2/2^{\ell(k)}.$$

It follows that B's advantage,

$$\begin{aligned} \epsilon'(k) &= \Pr[B^{\mathcal{F}_k}(1^k) = 1] - \Pr[B^{\mathcal{F}unc(k,k)}(1^k) = 1] \\ &\geq 1/2 + \epsilon(k)/2 - (1/2 + q(k)^2/2^{\ell(k)}) \\ &\geq \epsilon(k)/2 - q(k)^2/2^{\ell(k)}. \end{aligned}$$

Since $\mathcal{F}$ is a PRPF, $\epsilon'(k)$ is negligible. The number of queries $q(k)$ must be polynomial because of the running time of A is polynomial in k and since $\ell(k)$ is $k^{\Omega(1)}$ so $\epsilon(k) \leq \epsilon'(k) + q(k)^2/2^{\ell(k)}$ is negligible as required. □

There is one important thing to notice about the above proof that is a bit subtle. Although CBC_f^{IV} only makes sense as part of a symmetric encryption scheme if the function f is a permutation, it was OK to consider the behavior of the adversary A on the scheme even if f is not a permutation. The reason is that the definition of IND-CPA security of an encryption scheme $(\mathcal{K}, \mathcal{E}, D)$ depends on the properties an adversary A can determine based on access only to $\mathcal{K}$ and $\mathcal{E}$. If we wanted to try to extend this kind of argument to IND-CCA1 or IND-CCA2 security we would at least need to compare $\mathcal{F}$ to $\mathcal{P}erm(\ell(k), \ell(k))$ rather than $\mathcal{F}unc(\ell(k), \ell(k))$.

3.5 IND-CCA2 and NM-CPA security

For now, we won't formally define these notions of security but it is easy to see that the $CTR_{\mathcal{F}}$ scheme is not IND-CCA2 or NM-CPA secure even if $\mathcal{F}$ is a PRFF. The problem is the same as that for a one-time pad: One can observe that flipping a bit in a ciphertext block is equivalent to flipping the corresponding bit of the plaintext block.

The fact that $CBC_{\mathcal{F}}^{\mathcal{U}}$ is not IND-CCA2 or NM-CPA secure follows from the fact that by flipping a bit of $C_0 = IV$ one obtains an encryption of a message whose first block M_1 has its bit flipped.

(This is the only block where there is such a problem.) Doing something simple like applying F_K to the IV before doing the $\oplus$ doesn't help either. The problem in that case is that one could tell if $M_1 = 0^{\ell(k)}$.