Notes for Lecture 6

Scribed by Ian Haken, posted February 8, 2009

Summary

The encryption scheme we saw last time, based on pseudorandom functions, works and is CPA-secure, but it is not used in practice. A disadvantage of the scheme is that the length of the encryption is twice the length of the message being sent.

Today we see the "counter mode" generalization of that scheme, which has considerably smaller overhead for long messages, and see that this preserves CPA-security.

We then give the definition of *pseudorandom permutation*, which is a rigorous formalization of the notion of *block cipher* from applied cryptography, and see two ways of using block ciphers to perform encryption. One is totally insecure (ECB), the other (CBC) achieves CPA security.

1 The Randomized Counter Mode

Recall that a pseudorandom function is a function $F\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^m$ which looks approximately like a random function $R\colon \{0,1\}^m \to \{0,1\}^m$. With the encryption method from the previous lecture (in which the ciphertext is a random $r \in \{0,1\}^m$ followed by $F_K(r) \oplus M$) the encryption of a message is twice as long as the original message. We now define an encryption method which continues to use a pseudorandom function, but whose ciphertext overhead is marginal.

Suppose we have a pseudorandom function $F\colon \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^m$. We describe an encryption scheme that works for messages of variable length. We assume without loss of generality that the length of the message is a multiple of m, and we write a plaintext M of length cm as $M_1, \ldots, M_c$, a sequence of c blocks of length m.

- $Enc(K, M_1, \ldots, M_c)$:
  - pick a random $r \in \{0,1\}^m$
  - output $(r, F_K(r) \oplus M_1, F_K(r+1) \oplus M_2, \ldots, F_K(r+(c-1)) \oplus M_c)$
- $Dec(K, C_0, \ldots, C_c) := C_1 \oplus F_K(C_0), \ldots, C_c \oplus F_K(C_0 + (c-1))$

(When r is a binary string in $\{0,1\}^m$ and i is an integer, $r+i$ means the binary representation of the sum mod 2^m of r (seen as an integer) and i.)
Observe that the ciphertext length is $(c+1)m$ which is a negligable overhead when $c \gg m$.

Theorem 1 *Suppose F is a (t,ϵ)-secure pseudorandom function; then, when used to encrypt messages of length cm, the above scheme is $(t - O(cm), O(\epsilon + ct/2^m))$-CPA secure.*

Example 2 *Consider the values which these variables might take in the transmission of a large (e.g. $>$ 4GB) file. If we let $m = 128$, $t = 2^{60}$, $\epsilon = 2^{-60}$, $c = 2^{30}$, then we end up with an approximately $(2^{59}, 2^{-38})$-CPA secure transmission.*

PROOF: Recall the proof from last time in which we defined $\overline{Enc}(R,\cdot)$, where R is a truly random function. Given messages M, M' and a cryptanalytic algorithm T, we considered:

(a) $\mathbb{P}_K[T^{Enc(K,\cdot)}(Enc(K,M)) = 1]$

(b) $\mathbb{P}_R[T^{\overline{Enc}(R,\cdot)}(\overline{Enc}(R,M)) = 1]$

(c) $\mathbb{P}_R[T^{\overline{Enc}(R,\cdot)}(\overline{Enc}(R,M')) = 1]$

(d) $\mathbb{P}_K[T^{Enc(K,\cdot)}(Enc(K,M')) = 1]$

We were able to show in the previous proof that $|(a)-(b)| \leq \epsilon$, $|(c)-(d)| \leq \epsilon$, and $|(b)-(c)| \leq t/2^m$, thus showing that $|(a)-(d)| \leq 2\epsilon + t/2^m$. Our proof will follow similarly.

We will first show that for any M

$$\left|\mathbb{P}_K[T^{Enc(K,\cdot)}(Enc(K,M)) = 1] - \mathbb{P}_R[T^{\overline{Enc}(R,\cdot)}(\overline{Enc}(R,M)) = 1]\right| \leq \epsilon$$

hence showing $|(a)-(b)| \leq \epsilon$ and $|(c)-(d)| \leq \epsilon$. Suppose for a contradiction that this is not the case, i.e. $\exists M = (M_1, \ldots, M_c)$ and $\exists T$ where T is of complexity $\leq t - O(cm)$ such that

$$\left|\mathbb{P}_K[T^{Enc(K,\cdot)}(Enc(K,M)) = 1] - \mathbb{P}_R[T^{\overline{Enc}(R,\cdot)}(\overline{Enc}(R,M)) = 1]\right| > \epsilon$$

Define $T'^{O(\cdot)}(\cdot)$ as a program which simulates $T(O(M))$. (Note that T' has complexity $\leq t$). Noting that $T'^{Enc(K,\cdot)}() = T^{Enc(K,\cdot)}(Enc(K,M))$ and $T'^{\overline{Enc}(R,\cdot)}() = T^{\overline{Enc}(R,\cdot)}(\overline{Enc}(R,M))$, this program T' would be a counterexample to F being (t,ϵ)-secure.

Now we want to show that $\forall M = M_1, \ldots, M_c$, $\forall M' = M'_1, \ldots, M'_c$, and $\forall T$ such that the complexity of T is $\leq t - O(cm)$,

$$\left|\mathbb{P}_R[T^{\overline{Enc}(R,\cdot)}(\overline{Enc}(R,M))=1]-\mathbb{P}_R[T^{\overline{Enc}(R,\cdot)}(\overline{Enc}(R,M'))=1]\right|\leq 2ct/2^m$$

As in the previous proof, we consider the requests T may make to the oracle $\overline{Enc}(R,\cdot)$. The returned values from the oracle would be $r_k, R(r_k)\oplus M_1^k, R(r_k+1)\oplus M_2^k, \ldots, R(r_k+(c-1))\oplus M_c^k$, where k ranges between 1 and the number of requests to the oracle. Since T has complexity limited by t, we can assume $1\leq k\leq t$. As before, if none of the r_k+i overlap with $r+j$ (for $1\leq i,j\leq c$) then T only sees a random stream of bits from the oracle. Otherwise, if $r_k+i=r+j$ for some i,j, then T can recover, and hence distinguish, M_j and M'_j. Hence the probability of T distinguishing M, M' is ϵ plus the probability of a collision.

Note that the kth oracle request will have a collision with some $r+j$ iff $r-c<r_k\leq r+(c-1)$. If we have $r\leq r_k\leq r+(c-1)$ then obviously there is a collision, and otherwise $r-c<r_k<r$ so $r-1<r_k+(c-1)\leq r+(c-1)$ so there is a collision with $r_k+(c-1)$. If r_k is outside this range, then there is no way a collision can occur. Since r_k is chosen randomly from the space of 2^m, there is a $(2c-1)/2^m$ probability that the kth oracle request has a collision. Hence $2ct/2^m$ is an upper bound on the probability that there is a collision in at least one the oracle requests.

Combining these results, we see that $|(a)-(d)|\leq 2(\epsilon+ct/2^m)=O(\epsilon+ct/2^m)$, i.e.

$$\left|\mathbb{P}_K[T^{Enc(K,\cdot)}(Enc(K,M))=1]-\mathbb{P}_K[T^{Enc(K,\cdot)}(Enc(K,M'))=1]\right|=O(\epsilon+ct/2^m)$$

□

2 Pseudorandom Permutations

2.1 Some Motivation

Suppose the message stream has known messages, such as a protocol which always has a common header. For example, suppose Eve knows that Bob is sending an email to Alice, and that the first block of the message M_1 is the sender's email. That is, suppose Eve knows that M_1 ="bob@cs.berkeley.edu". If Eve can insert or modify messages on the channel, then upon seeing the ciphertext $C_0,\ldots,C_c$ she could then send to Alice the stream $C_0, C_1\oplus$ "bob@cs.berkeley.edu" $\oplus$ "eve@cs.berkeley.edu" $, C_2, \ldots, C_c$. The result is that the message received by Alice would appear to be sent from "eve@cs.berkeley.edu", but remain otherwise unchanged.

2.2 Definition

Denote by $\mathcal{P}_n$ the set of permutations $P\colon \{0,1\}^n\to\{0,1\}^n$.

Definition 3 *A pair of functions* $F\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, $I\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ *is a* (t,ϵ)*-secure pseudorandom permutation if:*

- *For every* $r \in \{0,1\}^k$*, the functions* $F_r(\cdot)$ *and* $I_r(\cdot)$ *are permutations (i.e. bijections) and are inverses of each other.*
- *For every oracle algorithm* T *that has complexity at most* t

$$\left| \mathbb{P}_K[T^{F_K, I_K}() = 1] - \mathbb{P}_{P \in \mathcal{P}_n}[T^{P, P^{-1}}() = 1] \right| \leq \epsilon$$

That is, to any algorithm T that doesn't know K, the functions F_K, I_K look like a random permutation and its inverse.

In applied cryptography literature, pseudorandom permutations are called *block ciphers*.

How do we construct pseudorandom permutations? There are a number of block cipher proposals, including the AES standard, that have been studied extensively and are considered safe for the time being. We shall prove later that any construction of pseudorandom functions can be turned into a construction of pseudorandom permutations; also, every construction of pseudorandom generators can be turned into a pseudorandom function, and every one-way function can be used to construct a pseudorandom generator. Ultimately, this will mean that it is possible to construct a block cipher whose security relies, for example, on the hardness of factoring random integers. Such a construction, however, would not be practical.

3 Encryption Using Pseudorandom Permutations

Here are two ways of using Pseudorandom Functions and Permutations to perform encryption. Both are used in practice.

3.1 ECB Mode

The Electronic Code-Book mode of encryption works as follows

- $Enc(K, M) := F_K(M)$
- $Dec(K, M) := I_K(M)$

Exercise 1 *Show that ECB is message-indistinguishable for one-time encryption but not for two encryptions.*

3.2 CBC Mode

In its simplest instantiation the Cipher Block-Chaining mode works as follows:

- $Enc(K, M)$: pick a random string $r \in \{0,1\}^n$, output $(r, F_K(r \oplus M))$
- $Dec(K, (C_0, C_1)) := C_0 \oplus I_K(C_1)$

Note that this similar to (but a bit different from) the scheme based on pseudorandom functions that we saw last time. In CBC, we take advantage of the fact that F_K is now a permutation that is efficiently invertible given the secret key, and so we are allowed to put the $\oplus M$ inside the computation of F_K.

There is a generalization in which one can use the same random string to send several messages. (It requires synchronization and state information.)

- $Enc(K, M_1, \ldots, M_c)$:
  - pick a random string $C_0 \in \{0,1\}^n$
  - output $(C_0, C_1, \ldots, C_c)$ where $C_i := F_K(C_{i-1} \oplus M_i)$
- $Dec(K, C_0, C_1, \ldots, C_c) := M_1, \ldots, M_c$ where $M_i := I_K(C_i) \oplus C_{i-1}$

Exercise 2 *This mode achieves CPA security.*

Note that CBC overcomes the above problem in which Eve knows a particular block of the message being sent, for if Eve modified C_1 in the encryption that Bob was sending to Alice (as in the example above) then the change would be noticeable because $C_2, \ldots, C_c$ would not decrypt correctly.