ICS 201: Cryptography and Communication Security 10/22/2007

Homework 2

Due Monday, Oct 29, in class

Problem 1 and 2

Do exercises 3.2 and 3.4.

Problem 3

Consider a stateful encryption scheme, e.g. a stream cipher. Here both the encryption and the decyrption algorithms take a state p as an additional input. The scheme also requires a state-update procedure, $p' \leftarrow \mathsf{Update}(p, m)$ for the encryptor (and a similar procedure for the decryptor). Let's assume that the initial state p is set to 0.

Consider a *Chosen Plaintext Attack* on a stateful encryption algorithm, with the attacker algorithm denoted A, running on input 1^n:

1. Key k is chosen by the key generation procedure on 1^n.
2. Let $p_1 = 0$ and $i = 1$. The following loop is repeated as long as A wants:

   A makes any *query* m_i to the encryption algorithm, which returns $c_i = Enc_k(m;\ p_i)$, i.e. the encryption of m_i under k given state p_i. The encryptor's state is updated, $p_{i+1} \leftarrow \mathsf{Update}(p_i, m_i)$, and we increase the i counter, $i \leftarrow i + 1$.
3. A outputs two messages $m^{(0)}, m^{(1)}$ of equal length (we call these *challenge messages*), bit b is chosen at random, and adversary is given $c = Enc_k(m^{(b)};\ p_i)$. The encryptor's state is updated, $p_{i+1} \leftarrow \mathsf{Update}(p_i, m^{(b)})$.
4. The adversary can now make more queries to the encryption algorithm, as in step 2.
5. When A is done with its queries, it outputs a bit b'.

Define the output of this experiment, denoted $\mathsf{CPA}_A(n)$, as 1 if $b = b'$ and as 0 otherwise. Note that $\mathsf{CPA}_A(n)$ is a random variable whose distribution is determined by randomness of A and the randomness of all procedures A interacts with, i.e. key generation, (possibly) encryption, and the choice of bit b.

We say that the stateful encryption scheme is CPA-secure (in the sense of indistinguishability) if for all efficient adversary algorithms A, there exists a negligible function ϵ s.t.

$$\Pr[\mathsf{CPA}_A(n) = 1] \leq 1/2 + \epsilon(n) \tag{1}$$

Compare this definition to definition 3.21 of Chosen-Plaintext-Attack Security in section 3.5: The only difference is that we consider a stateful encryption algorithm, and have update state at every encryption query. Note also that without steps 2,4, this becomes just the plain notion of encryption security (in the sense of indistinguishability), i.e. definition 3.8 in section 3.2.

Part 1

Show that if we implement a stateful encryption as a stream cipher built from a PRG, then this stateful encryption is CPA-secure as long as the total bit-length of all messages that the encryption encrypts is bounded by the output length of the PRG. In other words, let G be a PRG with expansion factor $l(n)$. Let the key generation pick a random n-bit string, and let encryption $Enc_k(m;\ p)$ output $m \oplus [G(k)]_p^{p+|m|-1}$, i.e. m xor-ed with the $|m|$ bits of $G(k)$ following the current pointer p. The update procedure $\mathsf{Update}(p, m)$ outputs $p + |m|$ (and same for c). Finally, the adversary has a limit on how many messages it can see encrypted: Their total bit-length must be at most $l(n)$.

Hint: The above claim generalizes the theorem theorem 3.16, and one way to prove it is to extend the proof of that theorem. You have to show that if G is a secure PRG then no efficient algorithm A can violate inequality (1), and you can do so by considering the following game *in which adversary A is involved: The game proceeds like the CPA experiment above, but in steps 2,3,4, the adversary is faced not with the real stream cipher, but with the algorithm* R *which does the following: At the beginning algorithm* R *picks a random $l(n)$-bit string r, and it encrypts m using procedure $Enc'_r(m;\ p) = m \oplus [r]_p^{p+|m|-1}$. In other words, R uses a true random string r instead of the pseudorandom string $G(k)$.*

Part 2

Consider an extension of the above CPA attack to a *Chosen (Plaintext and) Ciphertext* Attack on a stateful encryption. Namely, let the encryptor and decryptor have its own state, $p^{(e)}$ and $p^{(d)}$, respectively. Now in steps 2 and 4, the adversary A can makes both *encryption* queries m_i and *decryption* queries c_i. In response to an encryption query m_i the adversary gets $Enc_k(m_i;\ {p_i}^{(e)})$, and the encryptor state is updated, ${p_{i+1}}^{(e)} = \mathsf{Update}({p_i}^{(e)}, m_i)$, while in response to a decryption query c_i the adversary gets $Dec_k(c_i;\ {p_i}^{(d)})$, and the decryptor state is updated, ${p_{i+1}}^{(d)} = \mathsf{Update}({p_i}^{(d)}, c_i)$. The encryption queries are unrestricted, and so are the decryption queries in step 2, but in step 4 all decryption queries must differ from the challenge ciphertext c which A received in step 3. Note that this is like the CCA attack in definition 3.30, but here we define it for a stateful encryption scheme.

Show that a stream cipher built from a PRG is not secure under a CCA attack.

Hint: The attack is very simple...

Problem 4

Construct a PRG with expansion factor $l(n) = p(n) * n$ for any polynomial $p(n)$ given a PRF, and *prove* that your construction is a secure PRG assuming the underlying function is a PRF.

Hint: There are many possible constructions. In fact you can find examples of PRF→PRG construction embedded in several modes of operation of a block cipher.

Problems 5, 6, and 7

Do exercises 3.14 [showing CCA security of this construction is an optional, extra credit problem!], 3.16, and 3.22.