ICS 180: Introduction to Cryptography 6/12/2004

Homework 7

Due **Friday noon**, *6/18/2004*

1 PRGs

1.1 PRG warm-up

Recall the definition of a PRG. Consider the following attempt at constructing one: $G(x)$ outputs x concatenated with the parity bit of x, i.e. $G(x) = [x|b_{par}(x)]$, where $b_{par}(x)$ is the parity bit, i.e. it is 1 if x is even and 0 if x is odd. Is G a good PRG? (Prove or disprove.)

1.2 *Perfectly secure* PRG?

Remember perfectly secure encryption vs. computational notions of encryption security? Consider the following definition of a *perfect*, rather than computational, PRG: We say that a polynomial-time algorithm $G : \{0,1\}^k \to \{0,1\}^{k+1}$ is a perfect (one-bit-stretching) PRG if for *all algorithms* A we have:

$$Prob[A(y) = 1 \mid x \leftarrow \{0,1\}^k; y = G(x)] = Prob[A(y) = 1 \mid y \leftarrow \{0,1\}^{k+1}]$$

Note the two differences betwen this definition and the regular PRG definition: (1) The regular definition allows for a negligible difference between the above two probabilities, and (2) the regular definition asks this to hold not for all algorithms A but only for *probabilistic polynomial time* A's.

Show that "perfect PRGs" are too much to ask for, i.e. show that perfect PRGs do not exist. In other words, for any algorithm G show an algorithm A (not necessarily polytime) for which the above equation does not hold. What's your A's running time?

2 Encryption: Textbook vs. Indistinguishable Schemes

We show one clear flaw in plain (or "textbook") Rabin encryption, and we also show that an encryption scheme which is secure in the sense of indistinguishability is provably resistant to such flaws. Rabin's encryption is similar to RSA, and similar type of flaws, although technically slightly harder to show, can be shown for RSA, which is another argument why textbook RSA is not safe and why we need provably indistinguishable encryption schemes instead.

Here is a textbook Rabin public-key encryption: Recall the RSA function $RSA_{(n,e)} : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$, $RSA_{(n,e)}(x) = x^e \bmod n$ where n is the RSA modulus and e is for example 3. Rabin function is $Rabin_n : QR_n \to QR_n$, $Rabin_n(x) = x^2 \bmod n$ (recall that $QR_n \subset \mathbb{Z}_n^*$ is a set of squares modulo n). Inverting Rabin function means taking square roots $x = y^{1/2} \bmod n$, which is easy given the factorization of n. On the other hand, under the assumption that factoring is hard, one can prove (easily) that Rabin function is a TDP. Therefore it has a hard-core bit function, and therefore with some work we can construct

a provably indistinguishable encryption from it. However, in a *plain* Rabin encryption, assuming message $m \in QR_n$,[1] the ciphertext is simply $c = Rabin_n(m) = m^2 \bmod n$.

2.1 Textbook Rabin has a flaw

Suppose you want to encrypt two meesages, m and $m+1$, using textbook Rabin encryption. Show how an eavesdropper can recover m from these two ciphertexts.

2.2 Indstinguishable encryption withstands such attacks

Prove that if a public key encryption scheme $(KGen, Enc, Dec)$ is secure in the sense of indistinguishability of ciphertexts then such attack cannot be done. To show that, prove that an encryption scheme $Enc'(PK, m) = (Enc(PK, m), Enc(PK, m+1))$ is also secure in the sense of indistinguishability of ciphertexts.

Hint: The crux of the proof is in showing that while a single ciphertext $Enc(PK, m_1)$ *by itself* is assumed indistinguishable from $Enc(PK, m_1)$, and similarly $Enc(PK, (m_1 + 1))$ by itself is assumed indistinguishable from $Enc(PK, (m_2 + 1))$, one needs to argue why these two ciphertexts *together*, $(Enc(PK, m_1), Enc(PK, m_1+1))$, are indistinguishable from $(Enc(PK, m_2), Enc(PK, m_2+1))$. Note that these two ciphertexts are very much correlated since they use the same key and contain related messages, so the argument is not trivial. For example, if $|m| = |m+1| = k$ and Enc was a one time pad encryption using some fixed pad p, $|p| = k$, then the encryption is perfectly secure for single ciphertexts, but giving out two ciphertexts breaks the scheme immediately. Show that if Enc is an indistinguishable encryption this type of correlations do not matter. Use a hybrid argument to prove this.

3 Extending the PRF range

Recall problem 2 in problem set 5. Just like there, let $\{f_k\}$ be a PRF family s.t. if $|k| = \tau$ then f_k maps τ-bit inputs to τ-bit outputs.

3.1 PRF warm-up

Prove that for any constant $c \in \{0,1\}^\tau$, the following family $\{f'_k\}$ of functions $f'_k : \{0,1\}^\tau \to \{0,1\}^\tau$, is also a secure PRF family:

$$f'_k(x) = [\ f_{k'}(x)\ ] \text{ where } k' = f_k(c)$$

Hint: For any adversary A, consider how he does in telling conversations with O_{f_k} (for random key k) from conversations with a random function, and then ask what would it mean if A could tell conversations with an O_{f_k} oracle (for random k) from conversations with the $O_{f'_k}$ oracle (for random k).

[1] One can encode any message as a square by simply squaring the original message modulo n. There are some technicalities in unambiguously decrypting such message but we'll ignore this here.

3.2 PRF with polynomially-many blocks of output

We can use the above construction to extend the range of a PRF to any polynomial number of blocks. Namely, let $p(\tau)$ be some polynomial. Denote by $e(i)$ a binary encoding of a number $i = 0, 1, ..., p(\tau)$ as a τ-bit string, i.e., $e(0) = [00... ...00]$, $e(1) = [00... ...01]$, etc. Consider the following construction for a PRF family $\{f'_k\}$ where $f'_k : \{0,1\}^\tau \to \{0,1\}^{n*\tau}$ for $n = p(\tau)$:

$$f'_k(x) = [\ f_{k_1}(x) \mid f_{k_2}(x) \mid ... \mid f_{k_n}(x)] \text{ where } k_i = f_k(e(i))$$

Note that for $p(\tau) = 1$ and $c = [00... ...01]$, these are the same constructions. Use the previous part and a careful hybrid argument to prove this.

4 Message Authentication

Let's define a Message Authentication Scheme [MAC]. We say that a triple of PPT algorithms $(MGen, Tag, Ver)$ is a MAC if they satisfy the following conditions. First the *syntactic requirements*: (1) $MGen(1^\tau)$ on input a security parameter generates a symmetric key k of length polynomial in τ; (2) $Tag(k, m)$ on inputs a key k and a message m generates tag t; (3) $Ver(k, m, t)$ given key k, message m and a purported tag t says 1/0 depending if the tag is valid or not. The *security requirement* is the following: The triple of algorithms constitutes a secure MAC if for every PPT algorithm A involved in the following game with the MAC scheme, the probability of A's success, defined in step 4, is negligible:

1. A gets only the security parameter as an input
2. Algorithm $MGen(1^\tau)$ is run and the resulting key k is given to an oracle O_{MAC}.
3. Adversary A asks the O_{MAC} oracle for mac's on any message m_i he wants. The oracle returns $t_i = Tag(k, m_i)$. The adversary can repeat such querying of the oracle on any number of m_i's he wants.
4. Finally, A outputs a pair (m', t'), s.t. $m' \neq m_i$ for all the above m_i's. We say that A *succeeds* in an attack against this MAC scheme if $Ver(k, m', t') = 1$, i.e. if t' is indeed a proper message authentication code on m' under key k.

4.1 PRF-based MAC scheme for short messages

Show that for messages $|m| \leq \tau$ a simple use of a PRF makes a secure MAC scheme. Namely, show that if $\{f_k\}$ is a PRF family as in problem 3, $MGen$ picks a key $k \leftarrow \{0,1\}^\tau$, $Tag(k, m) = f_k(m)$ and $Ver(k, m, t)$ outputs 1 if $t = f_k(m)$, then $(MGen, Tag, Ver)$ form a secure MAC scheme.

4.2 Insecure MAC constructions for general messages

Let $\{f_k\}$ be a PRF family as in problem 3 and let encoding $e(\cdot)$ be like in problem 3.2. Show that the following PRF-based MAC constructions are all insecure. The $MGen$ algorithm always picks key $k \leftarrow \{0,1\}^\tau$. In each case we break up the message m into n τ-bit message blocks $m = [m_1, ..., m_n]$, where the last one, if its length is $k < \tau$, is appended with 1 and padded with a string of $(\tau - k - 1)$ zeroes, which ensures an unambiguous encoding (if the

number of bits in m is a multiple of τ, we append the whole τ-bit block [100... ...00] to m). The verification algorithm is the obvious thing corresponding to the Tag algorithm:

1. $Tag_1(k, [m_1, m_2, ..., m_n]) = [f_k(m_1) \oplus ... \oplus f_k(m_n)]$
2. $Tag_2(k, [m_1, m_2, ..., m_n]) = [f_{k_1}(m_1) \oplus ... \oplus f_{k_n}(m_n)]$ where $k_i = f_k(e(i))$
3. $Tag_3(k, [m_1, m_2, ..., m_n]) = t_n$ where $t_{i+1} = f_{k_i}(m_{i+1} \oplus t_i)$, $k_i = f_k(e(i))$, and $t_0 =$ [00... ...00]. [This one is the trickiest...][2]

5 Authenticated Encryption

Let $(KGen, Enc, Dec)$ be a symmetric encryptions scheme secure in the sense of indistinguishability against a Chosen Message Attack [CMA] (the adversary can ask for encryptions on any message he wants and then he still fails to distinguish ciphertexts of any two test messages of his choice). Let $(MGen, Tag, Ver)$ be a secure MAC scheme as defined in the previous problem. In practice, one wants communication to be both secret and authenticated. Consider the following algorithms for pair-wise communication which combine authentication with encryption. In each case first each party gets key (k_E, k_M) generated as $k_E \leftarrow KGen(1^\tau)$ and $k_M \leftarrow MGen(1^\tau)$:

1. "Encrypt-and-Authenticate" (this is a method used by SSH): The sender sends $(Enc_{k_E}(m), Tag_{k_M}(m))$. The receiver decrypts m and verifies it against the tag t.
2. "Authenticate-then-Encrypt" (this is what SSL does): The sender sends $Enc_{k_E}(m, Tag_{k_M}(m))$. The receiver decrypts (m, t) and verifies $Tag(k_M, m, t)$.
3. "Encrypt-then-Authenticate" (this is what IPSEC does): The sender computes $c = Enc_{k_E}(m)$ and sends $(c, Tag_{k_M}(c))$. The receiver verifies the ciphertext c against the tag and decrypts it if verification passes.

In each case, say whether this communication scheme is (1) a secure CMA encryption, and (2) whether it is a secure MAC scheme. Prove your answers (don't have to give full formalism, just give a counterexample/attack or a reasonable justification of why the proof of security should hold).

Hint: To examplify what we want above, let's see how to show that a CMA-secure encryption by itself, i.e. if the sender just sends $c = Enc_{k_E}(m)$, is not a good a MAC in general. One can show this by taking the PRF-based symmetric CMA-secure encryption scheme from class where $Enc_{k_E}(m) = (r, f_{k_E}(r) \oplus m)$ for random r in the domain of the PRF instance f_{k_E}. This procedure fails to provide a secure MAC because an adversary A, after querying the MAC oracle on any message m gets an answer, which a MAC-adversary interprets as a tag, $(t_1, t_2) = (r, f_{k_E}(r) \oplus m)$. Adversary A can then output a MAC forgery (t', m') where $t' = (t_1, t_2 \oplus (m \oplus m'))$, and you can check that t' is a valid tag on m'. Conclusion: CMA-secure encryption alone fails as a MAC.

[2]Insecurity of all these methods shows that making a secure MAC scheme from a PRF is indeed tricky. However, it's indeed possible (and done in practice) to cleverly chain block ciphers (which can be modeled as PRFs/PRPs) to get a MAC, and the secure construction is basically $Tag_4(k, m) = Tag_3(k, m')$ where m' is m *pre-pended* with the encoding of the length of m. You can find secure MAC constructions in lecture notes linked on our class website.