COS 433 — Cryptography — Homework 8.

Boaz Barak

Total of 120 points. Due April 9, 2010.

Exercise 1 (Random oracle security of hash and sign trapdoor - 30 points)**.** Recall that in class we considered the signature scheme that given a trapdoor function $f : \{0,1\}^n \to \{0,1\}^n$ and a hash function $H : \{0,1\}^m \to \{0,1\}^n$, signs $x \in \{0,1\}^m$ with key f^{-1} as $f^{-1}(H(x))$. We verify the signature t on x by checking that $f(t) = H(x)$. Prove that for every polynomial-time A, if H is chosen as a random function, and A is given the key f, black-box access to H, and the signing function $x \mapsto f^{-1}(H(x))$ then A wins the CMA game with at most negligible probability. (Where A wins the CMA game if it outputs (x^*, t^*) such that $t^* = f(H(x^*))$ but x^* is not one of the queries made by A to the signing function.)

Exercise 2 (Non malleability of CCA secure schemes - 30 points)**.** An attractive way to perform a bidding is the following: the seller publishes a public key e. Each buyer sends through the net the encryption $\mathsf{E}_e(x)$ of its bid, and then the seller will decrypt all of these and award the product to the highest bidder.

One aspect of security we need from $\mathsf{E}(\cdot)$ is that given an encryption $\mathsf{E}_e(x)$, it will be hard for someone not knowing x to come up with $\mathsf{E}_e(x+1)$ (otherwise bidder B could always take the bid of bidder A and make into a bid that is one dollar higher). You'll show that this property is also related to CCA security:

1. Show a CPA-secure public key encryption such that there is an algorithm that given e and a ciphertext $y = \mathsf{E}_e(x)$, converts y into a ciphertext y' that decrypts to $x+1$. (If it makes your life easier, you can make the algorithm work only if x is, say, a multiple of 100.)

2. Show that if E is CCA secure then there is no such algorithm, in the following sense: that if M is any polynomial time algorithm, then

$$\Pr_{\substack{(e,d)\leftarrow\mathsf{Gen}(1^n)\\ X\leftarrow_{\mathrm{R}}\{1,10^6\}}}[\mathsf{D}_d(M(e,\mathsf{E}_e(x))) = x+1] < 10^{-6} + n^{-\omega(1)}$$

Exercise 3 (Authenticated key exchange - 60 points)**.** Consider a key exchange protocol where the client has the public keys of a server, chooses a key $k \leftarrow_{\mathrm{R}} \{0,1\}^n$ for a private key scheme, interacts with the server, and at the end decides whether or not to accept the key as valid. For simplicity we restrict ourselves to two-message protocols (one message from client to server and one message from server to client). Consider the following attack on such protocols: (In this attack the adversary completely controls the network between the client and server, so that all messages transmitted between them go through the adversary.)

1. Client sends the first message to the adversary.

2. Adversary gets a polynomial number of interactions with the server, in each such interaction the adversary sends a message to the server. The server interprets the message as a first-message from some client, and it either accepts a key k as a result of this message and outputs the second message of the protocol or it outputs "invalid". If the server accepted the key k, it also outputs $\mathsf{E}_k^{\mathsf{priv,cca}}(0^n)$. The adversary gets the outputs of the server.

3. Adversary sends a message to the client.

4. If the client accepts the message and obtained a key k, then it chooses $b \leftarrow_{\mathrm{R}} \{0,1\}$, and does the following. If it accepted the key k then the client outputs an encryption $\mathsf{E}_k^{\mathsf{priv,cca}}(0^n)$ if $b = 0$, and $\mathsf{E}_k^{\mathsf{priv,cca}}(1^n)$ if $b = 1$. Otherwise it outputs "invalid".

5. The adversary outputs $b' \in \{0,1\}$. We say the adversary is *successful* if both (i) the client accepted the key and (ii) $b' = b$.

We say the protocol is *secure* if the probability the adversary succeeds in this attack is at most $\frac{1}{2} + n^{-\omega(1)}$.

Notation: We denote by $(\mathsf{Sign}, \mathsf{Ver})$ a secure signature scheme. We denote by $\mathsf{E}^{\mathsf{pub,cca}}$ a CCA secure public key encryption scheme, by $\mathsf{E}^{\mathsf{pub,cpa}}$ a CPA secure public key encryption scheme, and by $\mathsf{E}^{\mathsf{priv,cca}}$ a CCA secure private key encryption scheme. The protocol is secure if it is secure for *every* suitable choice of the underlying schemes. In all cases we denote by e and by v the public encryption key and verification key of the server, and assume that the client knows them.

For each of the following protocols, either prove that it is secure (for *every* suitable choice of the schemes) or give an example showing it is insecure (for *some* choice of the schemes).

Protocol 1:

- Client chooses $k \leftarrow_{\mathrm{R}} \{0,1\}^n$ and $m \leftarrow_{\mathrm{R}} \{0,1\}^n$ and sends to server $\mathsf{E}_e^{\mathsf{pub,cpa}}(k \circ m)$. ($\circ$ denotes string concatenation.)

- Server decrypts ciphertext to get k, m, accepts the key k, and sends to client $m, \mathsf{Sign}_s(m)$ (if ciphertext is invalid then server sends "invalid").

- Clients verifies m is the same string it sent before, verifies signature and if it passes verification, it considers the key k as valid.

Protocol 2: Same as Protocol 1 but with $\mathsf{E}^{\mathsf{pub,cca}}$ instead of $\mathsf{E}^{\mathsf{pub,cpa}}$.

Protocol 3:

- Client chooses $k \leftarrow_{\mathrm{R}} \{0,1\}^n$ and sends to server $y = \mathsf{E}_e^{\mathsf{pub,cpa}}(k)$.

- Server decrypts ciphertext to get k, chooses $m \leftarrow_{\mathrm{R}} \{0,1\}^n$ at random and sends to client y,m and $\mathsf{Sign}_s(y \circ m)$ (if ciphertext is invalid then server sends "invalid").

- Client checks y is the same message it sent before, verifies signature and if it passes verification, it considers the key k as valid.