COS 433 — Cryptography — Homework 7.

Boaz Barak

Total of 120 points. Due November 15th, 2007.

Exercise 1 (30 points)**.** Consider the following identification problem: there's a "box" that controls access to some resource (e.g., a door) and authorized people are given some secret information that enables them to use the resource, but unauthorized people cannot do so, even if the know the contents of the box.

1. Consider the weakest security definition of this problem where the only attack we're considering is an adversary that "listens in" on conversations between honest users and the box). Construct a non-interactive (i.e., a protocol consisting of a single message from the user to the box) protocol solving the identification problem and prove its security.

2. Construct a non-interactive identification protocol that remains secure under the stronger attack where an adversary may open up the box and see the data it contains, listen in on other conversations and also construct "fake boxes" and have the authorized people talk to these fake bozes. Prove the security for your protocol. You may assume that all parties have access to a perfectly synchronized global clock.

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 particular show that if M is any polynomial time algorithm, and X is a set of possible numbers x, then

$$\Pr_{(e,d)\leftarrow\mathsf{Gen}(1^n)}[\mathsf{D}_d(M(e,\mathsf{E}_e(x))) = x+1] < \tfrac{1}{|X|} + n^{-\omega(1)}$$

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:

- Client sends the first message to the adversary.
- Adversary gets a polynomial number of interactions with the server.
- Adversary sends a message to the client.
- The client chooses $b \leftarrow_{\mathrm{R}} \{1,2\}$. If it accepts the message and obtained a key k *and* $b = 1$ then it sends k to the adversary. Otherwise, (either $b = 2$ or it did not accept the message) it sends a random string $k' \leftarrow_{\mathrm{R}} \{0,1\}^n$ to the adversary.
- The adversary outputs $b' \in \{1,2\}$. We say the adversary is successful if $b' = b$.

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

For each of the following protocols, either prove that it is secure or give an example showing it is insecure. **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 $\mathsf{E}^{\mathsf{priv,cpa}}$ a CPA secure private key encryption scheme. The protocol is secure if it is secure for any 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.

Protocol 1:

- Client chooses $k \leftarrow_{\mathrm{R}} \{0,1\}^n$ and $m \leftarrow_{\mathrm{R}} \{0,1\}^n$ and sends to server $\mathsf{E}_e^{\mathsf{pub,cpa}}(k \circ m)$.
- Server decrypts ciphertext to get k, m and sends to client $m, \mathsf{Sign}_s(m)$ (if ciphertext is invalid then server sends "invalid".
- Clients verifies signature and if it passes verification, it considers the key k as valid. (It will use m as a label of the key in future conversations with the server)

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

Protocol 3:

- Client chooses $k \leftarrow_{\mathrm{R}} \{0,1\}^n$, $k' \leftarrow_{\mathrm{R}} \{0,1\}^n$ and $m \leftarrow_{\mathrm{R}} \{0,1\}^n$ and sends to server $\mathsf{E}_e^{\mathsf{pub,cpa}}(k \circ k' \circ m)$.
- Server decrypts ciphertext to get k, k', m and sends to client $y = \mathsf{E}_{k'}^{\mathsf{priv,cpa}}(m)$ and $\mathsf{Sign}_s(y)$ (if ciphertext is invalid then server sends "invalid".
- Clients verifies signature and if it passes verification and y decrypts with k' to the value m, it considers the key k as valid.

Protocol 4:

- 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".
- Clients verifies signature and if it passes verification, it considers the key k as valid.