Homework 7: Authenticated Key Exchange, CCA for public key, zero knowledge

Total of 175 points

1. (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 $E_e(x)$ 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 to achieve this is that given an encryption $E_e(x)$, it will be hard for someone not knowing x to come up with $E_e(x+1)$ (otherwise bidder B could always take the bid of bidder A and make into a bid that is one dollar higher).

    a. (15 points) Show a CPA-secure public key encryption (G, E, D) on messages in $\{0,1\}^\ell$ (interpreted as numbers in $[0, 2^\ell - 1]$) such that there is an algorithm that given e and a ciphertext $c = E_e(x)$ for $x < 2^\ell$, outputs a ciphertext c' that decrypts to $x+1$. (If it makes your life easier, you can make the algorithm work only if x is a multiple of 2^{10}.)

    b. (15 points) Show that if (G, E, D) is CCA secure there is no such algorithm in the following sense: if M is any polynomial time algorithm then the probability over $x \leftarrow_R [0, 2^\ell - 1]$ and $(e, d) = G(1^n)$ thet $D_d(M(E_e(x))) = x + 1$ is negligible.

2. (60 points) Consider a key exchange protocol where the client has the public keys of a server, chooses a key $k \leftarrow_R \{0,1\}^n$ 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 $E_k^{priv,cca}(0^n)$. 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 $b \leftarrow_R \{0,1\}$, and does the following. If it accepted the key k then the client outputs an encryption $E_k^{priv,cca}(0^n)$ if $b = 0$, and $E_k^{priv,cca}(1^n)$ if $b = 1$. Otherwise it outputs `invalid`.
5. The adversary outputs $b' \in \{0,1\}$. We say the adversary *wins* 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 $1/2 + negl(n)$

Notation: We denote by (S, V) a secure signature scheme. We denote by $E^{pub,cca}$ a CCA secure public key encryption scheme, by $E^{pub,cpa}$ a CPA secure public key encryption scheme, and by $E^{priv,cca}$ 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: (20 points)

1. Client chooses $k \leftarrow_R \{0,1\}^n$ and $m \leftarrow_R \{0,1\}^n$ and sends to server $E_e^{pub,cpa}(k\|m)$.
2. Server decrypts ciphertext to get k, m, accepts the key k, and sends to client $m, S_s(m)$ (if ciphertext is invalid then server sends `invalid`).
3. Client 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: (20 points) Same as Protocol 1 but with $E^{pub,cca}$ instead of $E^{pub,cpa}$.

Protocol 3: (20 points)

1. Client chooses $k \leftarrow_R \{0,1\}^n$ and sends to server $y = E_e^{pub,cpa}(k)$.

2. Server decrypts ciphertext to get k, chooses $m \leftarrow_R \{0,1\}^n$ at random and sends to client y,m and $S_s(y\|m)$ (if ciphertext is invalid then server sends `invalid`).

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

3. (*zero knowledge is trivial for easy languages*, 15 points) We say that $L \subseteq \{0,1\}^*$ is decidable in probabilistic polynomial time (in complexity parlance $L \subseteq BPP$) if there is a probabilistic polynomial time algorithm A such that if $x \in L$ then $\Pr[A(x) = 1] > 2/3$ and if $x \notin L$ then $\Pr[A(x) = 1] < 1/3$. Show that if L is decidable in probabilistic polynomial time then it has a trivial zero knowledge proof system (P, V) in which the prover sends no messages to the verifier.

4. (*Interaction is necessary for non-trivial zero knowledge*, 15 points) Let $L \subseteq \{0,1\}^*$. We say that L has a *non interactive* zero knowledge proof system if there it has a zero knowledge proof system (P, V) consisting solely of P sending a single message to the verifier. Show that if there is a non-interactive zero knowledge proof system for L then it is decidable in probabilistic polynomial time.[1]

5. (*Randomness is necessary for non-trivial zero knowledge*, 15 points) Show that if a language L has a proof system where the verifier is deterministic (i.e., uses no randomness) then L is decidable by a probabilistic polynomial time algorithm that may use some non uniform "hardwired constants" of $poly(n)$ size.

6. (40 points) For some prime $q = poly(n)$ and $m \times n$ matrix A over $\mathbb{Z}_q$, we say that A is "spread out" if for every $y \in \mathbb{Z}_q^n$, the probability over random $w \in \{0,1\}^m$ that $w^\top A = y$ is at most $10 \cdot 2^{-n}$.

   a. (10 points) Prove that if n is sufficiently large, A is a random matrix with entries independently chosen from $\mathbb{Z}_q$ and $m > 1000n \log q$ then with probability at least 0.99 A is "spread out". See footnote for hint[2]

   b. (10 points) Prove that if A has the form $(A'|y)$ where A' is an $m \times n-1$ matrix and $y \in \mathbb{Z}_m^q$ has the property that $y = A'x + e$ for some $x \in \mathbb{Z}_q^{n-1}$ and e satisfying $\sum_{i=1}^m |e_i| < q/(100n)$ then A' is *not* spread out.

   c. (20 points) Give a proof system (P, V) (with a potentially inefficient prover) such that on input A, if A is spread out then he can convince

[1] Because of this negative result, the standard definition in cryptography of a *non interactive zero knowledge proof system (NIZK)* is different and assumes some global trusted public parameters (known as a "common reference string"). This allows to bypass the impossiblity result.

[2] You can use the Leftover Hash Lemma as was stated in the lecture on lattices.

the verifier that it does *not* have the form $A = (A'|y)$ as in item b. above.

d. (no points, don't write - just think about it) Can you come up with a *zero knowledge* proof system by which a prover could prove to the verifier in zero knowledge that if A has the form above then it is at least not spread out? One fact you might want to use is that if e has small magnitude and we pick a random e' of much larger (but still not too large) magnitude, then the distributions $\{e'\}$ and $\{e + e'\}$ are at least weakly indistinguishable from one another.