CS195-7: Introduction to Cryptography DUE: December 17, 2002

Final Exam

Instructor: Anna Lysyanskaya

Problem 1

Let (G, E, D) be a public-key cryptosystem. Suppose Alice has a key pair (PK_A, SK_A) for this cryptosystem, obtained by running algorithm G. One day, Alice goes on vacation and wants her friend Bob to read her mail while she is away. So she needs to give Bob her secret key SK_A. Bob lives in another city, and so Alice has to send her secret key to him by e-mail. Bob's public key PK_B is known to her, so she generates the ciphertext $c_{SK} \leftarrow E(PK_B, SK_A)$ and sends c_{SK} to Bob.

(a) Explain what it means for Alice's mail to be semantically secure even after the eavesdropper Eve gets hold of the values (PK_A, PK_B, c_{SK}). (Since we are only after *semantic* security, Eve is passive, i.e., she does not ask Alice or Bob to actually decrypt any ciphertexts.) Give both an English explanation, and a formal definition.

(b) Show that if (G, E, D) is semantically secure, then Alice's mail remains semantically secure.

(c) Explain what it means for Alice's mail to be chosen-ciphertext secure even after the eavesdropper Eve gets hold of the values (PK_A, PK_B, c_{SK}) *and* is allowed access to the appropriate decryption algorithms. Does it follow that if (G, E, D) is secure against chosen ciphertext attack, then in this altered scenario, Alice's mail will also remain secure against the chosen ciphertext attack? You do not have to give a formal proof, but argue why your proof for part (b) works or does not work here.

(d) Suppose that, instead of encrypting her SK_A under Bob's public key, Alice encrypted it under her own public key PK_A (I have no idea why she would want to do that!), and so $c_{SK} \leftarrow E(PK_A, SK_A)$. Does it follow that, if (G, E, D) is a semantically secure cryptosystem, Alice's mail will remain secure after the adversary gets hold of (PK_A, c_{SK})? Does it follow that her mail becomes insecure?

Problem 2

Recall the ElGamal cryptosystem:

Key generation On input 1^k, the key generation algorithm outputs the values $(PK = (p, q, g, h), SK = x)$, where (1) $p = aq + 1$ is a k-bit prime number; (2) q is a prime number of $O(k)$ bits (usually $p = 2q+1$, but it is unknown whether there is an infinite number of such primes); (3) g is a generator of a subgroup of $\mathbb{Z}_p^*$ of order q, denote this subgroup by $G = \langle g \rangle$; (4) $x \leftarrow \mathbb{Z}_q$, $h = g^x \bmod p$.

Encryption To encrypt a message $m \in G$, pick $r \leftarrow \mathbb{Z}_q$, and output the ciphertext $c = (g^r, h^r m)$.

Decryption To decrypt a ciphertext $c = (a, b)$, output $m = b/a^x$.

In class, we saw that this cryptosystem is semantically secure under the decisional Diffie-Hellman assumption, i.e., the assumption that the following two experiments produce indistinguishable distributions:

Experiment DH On input 1^k, choose p, q, g as described above. Pick random $x \leftarrow \mathbb{Z}_q$, $y \leftarrow \mathbb{Z}_q$, and let $X = g^x$, $Y = g^y$, $Z = g^{xy}$. Output (p, q, g, X, Y, Z).

Experiment R On input 1^k, choose p, q, g as described above. Pick random $x \leftarrow \mathbb{Z}_q$, $y \leftarrow \mathbb{Z}_q$, $z \leftarrow \mathbb{Z}_q$, and let $X = g^x$, $Y = g^y$, $Z = g^z$. Output (p, q, g, X, Y, Z).

Give a chosen-ciphertext attack against this cryptosystem.

Problem 3

The Schnorr signature scheme is secure in the random oracle model under the assumption that the discrete logarithm problem is hard. It works as follows:

Key generation Pick a group G: choose a k-bit prime $p = aq + 1$, where q is a prime number of length $O(k)$, and produce (p, q, g), where $g \in \mathbb{Z}_p^*$ has order q. Let $G = \langle g \rangle$. Pick a hash function $H : \{0,1\}^* \mapsto \mathbb{Z}_q$. Pick the secret key $x \leftarrow \mathbb{Z}_q$. Let $h = g^x$. Output the public key (p, q, g, h).

Signing In order to sign a message m, choose a random $r \leftarrow \mathbb{Z}_q$. Compute the following:

$$\begin{aligned} A &= g^r \bmod p \\ c &= H(m \circ A) \\ s &= r + xc \bmod q \end{aligned}$$

and output the signature (A, s).

Verification In order to verify a signature (A, s) on message m, compute $c = H(m \circ A)$, and check that $g^s = Ah^c$.

Here, we will prove this signature scheme secure.

(a) Prove that the Schnorr signature scheme is non-trivial, i.e., that any signature generated by the signing algorithm will pass the verification test.

(b) Show that one must "know" x in order to compute a signature correctly, in the following sense: if for the same A, one can give the right values s_1 and s_2 for two *different* values c_1 and c_2 of $H(m \circ A)$, then x can be derived. More precisely, design an efficient algorithm that, given (A, c_1, c_2, s_1, s_2), such that $c_1 \neq c_2$, and $g^{s_1} = Ah^{c_1}$, while $g^{s_2} = Ah^{c_2}$, outputs x such that $h = g^x$.

(c) As always, security has to be proved by reduction. The reduction has access to an adversary that forges Schnorr signatures with probability $P(k)$. The reduction receives as input the group G represented by (p, q, g), and the value $h = g^x$. The goal of the reduction is to compute x.

The reduction must interact with the adversary, i.e., set up a public key of a signature scheme and answer the adversary's signature queries. In addition, since this proof of security is in the random oracle model, the reduction is allowed to implement the ideal hash function H for the adversary, i.e., answer the adversary's hash queries.

The reduction sets up the public key as follows $PK = (p, q, g, h, H)$. In order to answer a signature query on message m, the reduction proceeds as follows: pick a random $c \leftarrow \mathbb{Z}_q$ and $s \leftarrow \mathbb{Z}_q$. Let $A = g^s/h^c$. Fix the random oracle such that $H(m \circ A) = c$. In order to answer a hash query, the reduction just outputs a random string.

Eventually, the adversary outputs a forged signature (A, s) on message m. Since this is a valid forgery, $g^s = Ah^c$, where c is the answer H gave to the query $(m \circ A)$. At this point, the reduction will *reset* the adversary to the point in time in which it gave c in response to $(m \circ A)$, and respond with a different random $c' \leftarrow \mathbb{Z}_q$, and run the adversary from that point on. If the adversary outputs a forgery on the same message m and for the same value A, then by part (b), our reduction can compute x, and therefore succeeds. Otherwise, our reduction fails.

Show that if the success probability of the adversary is $P(k)$, and his running time is $T(k)$, then the reduction described above succeeds with probability at least $P^2(k)/T(k)$.