University of California — Berkeley
CS276: Cryptography
Professors Luca Trevisan and David Wagner

Handout PS1
February 12, 2002

Problem Set 1

This problem set is due in class on **Thursday, February 21.**

Problem 1. [Power of Simulator in Semantic Security]

For a public-key encryption system (G, E, D), consider the following notion of weak-(t, ϵ)-Semantic-Security;

- For every message distribution X, side information h, desired information f, and adversary A running in time $\leq t(n)$, there is a simulator S of *arbitrary* running time such that

$$\Pr_{(pk,sk)\leftarrow G, m\leftarrow X}[A(E(pk,m), pk, h(m)) = f(m)] \leq \Pr_{m\leftarrow X}[S(pk, h(m)) = f(m)] + \epsilon$$

Notice how the arbitrary running time of the adversary makes this definition weaker than the one seen in class. Then consider the following definition of strong-(t, ϵ)-Semantic Security.

- There is a polynomial p such that for every message distribution X, side information h, desired information f, and adversary A running in time $t_A(n) \leq t(n)$, there is a simulator S running in time $\leq t_A(n) + p(n)$ such that

$$\Pr_{(pk,sk)\leftarrow G, m\leftarrow X}[A(E(pk,m), pk, h(m)) = f(m)] \leq \Pr_{m\leftarrow X}[S(pk, h(m)) = f(m)] + \epsilon$$

Notice that this definition is stronger because the adversary has to run in time $t_A(n) + poly(n)$ instead of $poly(t_A(n))$, and typically we are interested in the case where $t_A()$ is super-polynomial.

Prove that the two definitions are equivalent and, specifically, if a system is weak-(t, ϵ)-semantically secure, then it is also strong-$(t, 2\epsilon)$-semantically secure.

Problem 2. [Weak Decryption Oracle]

Let's define a new notion, IND-CCA0, that is just like IND-CCA2, except that instead of being given access to an oracle D_{sk}, the adversary is given access to an oracle O defined as follows. On query (y, x), the oracle O returns "true" if $D_{sk}(y) = x$, else "false." (Just like in IND-CCA2, we forbid the adversary from querying its oracle on a pair (y, x) where

y is the ciphertext it was given as input, but the adversary can query O on anything else.) Obviously IND-CCA0 lies somewhere between IND-CPA and IND-CCA2.

Show a separation, i.e., show that if there is a public-key encryption system that satisfies IND-CPA, then there is also a public key encryption system that satisfies IND-CPA but not IND-CCA0.

Problem 3. [A Different way of Using a Hash Function]

Suppose we use Goldwasser-Micali-RSA to transmit network packets. Network packets have a cryptographic checksum in them; let's imagine that before encrypting a message m, we prepend an ideal hash of m (that is, $G(m)$, where $G()$ is a random oracle and, say, $G()$ maps strings of length n to strings of length n) to m.

Suppose that the adversary can issue a limited sort of chosen-ciphertext query: the adversary is given an oracle O and can submit ciphertext y to the oracle O. The oracle O will respond "true" if $D(sk, y)$ decrypts to something with a valid hash (i.e., something of the form (d, m) with $d = G(m)$), or "false" otherwise. In addition, the adversary is also given oracle access to the random oracle G (as always in the random oracle model). Thus, the notion of security is like that of IND-CPA, extended with oracles O and G.

Prove or disprove: GM-RSA is secure against this sort of attack.

> **Remark.** This problem is based on practical considerations. If we consider a recipient who decrypts a ciphertext and then does something useful with the result, then in practice quite frequently it will be possible to distinguish the case where the decryption was valid from the case where it was not, simply by examining the reaction of the recipient or by closely monitoring externally-observable behavior of the recipient.
>
> For example, in SSL, the recipient explicitly sends back an error message if the decryption is invalid. In IPSec, if the plaintext is a TCP/IP packet, as is common, then the recipient will respond with a TCP ACK packet if the decryption was valid, but will do nothing otherwise. In many systems, the recipient aborts early if the ciphertext is invalid and so may spend more time computing if the decryption is valid; thus, by observing timing differences we may be able to learn whether the decryption of a chosen ciphertext is valid. Therefore, in practice the existence of this sort of attack against an encryption scheme would be troubling. ■

Problem 4. [Proofs of Security for CCA2 Attacks]

If (G, F, I) is a trapdoor permutation with a hardcore bit $B()$, consider the encryption scheme $E(pk, x) = (F(pk, r), B(r) + x)$, where x is a one-bit plaintext and r is a random nonce chosen uniformly at random, independently of everything else, and anew for each encryption. We showed in class that $E()$ is IND-CPA secure if (G, F, I) is, using the following type of argument: if there is any IND-CPA attack on GL given just the public key and a ciphertext, then there is an attack that inverts F given just an output of F.

(a) Prove a corresponding result about $E()$'s security against chosen-ciphertext attacks: if there is any IND-CCA2 attack on $E()$ given just the public key pk, a ciphertext $c = (v, w)$, and a CCA2-style decryption oracle (restricted so that we can't query it on any ciphertext of the form $(v, \cdot)$), then there is an attack that inverts $F(pk, \cdot)$ given just pk, an output y of $F(pk, \cdot)$, and an inversion oracle that on query z returns $I(sk, z)$ (but restricted so that we can't query our inversion oracle on y).

(b) Is this a good basis for concluding that $E()$ is a good scheme to use in practice?

Problem 5. [Super-encryption]

In this problem, the goal is to show that super-encryption (i.e., repeated encryption) can't hurt security.

Let's consider the following scenario. You pick an IND-CPA secure encryption scheme (E, D, G), choose a public/private keypair $(pk, sk) \leftarrow G$, publish pk, and begin encrypting all packets leaving your machine under E_{pk}. Thus, the only thing an eavesdropper would be able to see is ciphertexts of the form $E_{pk}(x)$.

Unbeknownst to you, the University has also decided to encrypt all packets exiting the University's network. In particular, the University has independently picked some IND-CPA secure encryption scheme (E', D', G'), has independently chosen a public/private keypair $(pk', sk') \leftarrow G'$, has published pk', and the University's border routers take every outgoing packet they see and mechanically encrypt it under $E_{pk'}$.

Thus, though you didn't anticipate it, all packets you send out onto the Internet are double-encrypted: they have the form $E_{pk'}(E_{pk}(x))$. Show that the result is still secure against chosen-plaintext attacks (IND-CPA). First formalize the problem appropriately, state a security theorem, and then prove it.

> **Remark.** This problem is also motivated by many practical applications. What this sort of theorem is giving you is a nice modularity result: all you need to do to ensure security is to pick a good encryption scheme yourself, and what the University does with your ciphertext afterwards can't possibly hurt security. Modularity and composition are critical when building complex security-critical systems. ■