ITC8240 Cryptography. Exam. Test2

Urmas Pitsi, 7.jan.2021

1. Show that a cryptosystem, that is indistinguishable under chosen plaintext attack and is homomorphic with respect to multiplication, is not indistinguishable under adaptive chosen ciphertext attack.

Adversary A can follow these steps.
Adversary A:

1. receives PK (public key)
2. Wait
3. A: M0, M1 -> Environment
4. Receive C = Encr(PK, Mb), message: Mb
5. Select invertible element mod n
   let blinding factor be 2
   (2^e mod n) * C = (2^e mod n)(Mb^e mod n) = (2*Mb)^e mod n, this is by homomorphic property.
   Submit '(2*Mb)^e mod n' to the decryption oracle.
   Receive 2*Mb
   Calculate 2^-1 * 2 * Mb = Mb.
6. Return 1 if Mb=m1, otherwise return 0.

This gives Adversary A the advantage of 0.5 which is non-negligible.

2. Show that a cryptosystem that is indistinguishable under adaptive chosen ciphertext attack is indistinguishable under chosen plaintext attack. Hint: to show this, show that distinguishability under chosen plaintext attack implies distinguishability under adaptive chosen plaintext attack.

In IND-CCA2 the adversary has the access to a decryption oracle. So if a cryptosystem is IND-CCA2 and we remove the adversary's access to decryption oracle, the resulting cryptosystem remains IND-CPA.

IND-CPA can be modeled as following game between an adversary and a challenger:

1. The challenger generates a key pair PK, SK based on some security parameter k (e.g., a key size in bits), and publishes PK to the adversary. The challenger retains SK.
2. The adversary may perform a polynomially bounded number of encryptions or other operations.
3. Eventually, the adversary submits two distinct chosen plaintexts M0, M1 to the challenger.
4. The challenger selects a bit b in {0, 1} uniformly at random, and sends the challenge ciphertext C = E(PK, Mb) back to the adversary.
5. The adversary is free to perform any number of additional computations or encryptions. Finally, it outputs a guess for the value of b.

A cryptosystem is IND-CPA if every probabilistic polynomial time adversary has only a negligible "advantage" over random guessing.
Now let's assume a cryptosystem is not IND-CPA that means the adversary has non-negligible advantage over random guessing. If on top of that the adversary has the access to the decryption oracle in step 5, it would make the advantage even bigger.

3. Find a second pre-image of 83 given a hash function h : Z×Z → U(98) defined by h : x → 5^x mod 98 and a pre-image 9

Second pre-image = 51.
h(a) = 83 -> a = {9, 51, ...} as 5^9 mod 98 = 5^51 mod 98 = 83.

4. Show that all proper values of the RSA public exponent are odd.

$encryption\ c\ =\ m^e\ mod\ n$

$decryption\ D\ =\ c^d\ mod\ n$

By definition:

n = *p* * *q*, where *p* and *q* are prime numbers and Phi(n) = (p-1)*(q-1).

We know that exponents *e* and *d* are such that: $d \equiv e^{-1}\ mod\ \Phi(n)$, ie *d* is the multiplicative inverse of *e* mod Phi(n).

If *e* is even, then no *d* exists that makes an even number congruent to 1 mod even number.

5. Show that if there exists a poly-time algorithm to factor integers, then there exists a poly-time attack on RSA that can be used to derive a private exponent out of a public exponent and an RSA modulus.

$$encryption\ c\ =\ m^e\ mod\ n$$
$$decryption\ D\ =\ c^d\ mod\ n\ =\ m$$

where *c*=cipher text, *m*=message, *e* and *n* are publicly available numbers.
By definition:
n = *p* * *q*, where *p* and *q* are prime numbers and Phi(n) = (p-1)*(q-1).
We know that exponents *e* and *d* are such that: $d \equiv e^{-1}\ mod\ \Phi(n)$, ie *d* is the multiplicative inverse of *e* mod Phi(n).

Now suppose there exists a poly-time algorithm to factor integers, then we can factor *n* to obtain *p* and *q*. Then we can easily calculator Phi(n) which is (p-1)*(q-1).

By knowing public exponent *e* and Phi(n) we can derive private exponent *d* in poly-time as *d* is the multiplicative inverse of *e* mod Phi(n). We know that multiplicative inverse can be calculated in poly-time(using Extended Euclidean algorithm), ie finding d, given e and Phi(n):
$d \equiv e^{-1}\ mod\ \Phi(n)$.

6. You have a message m that you need to be signed by Alice. How do you obtain Alice's signature on m without Alice knowing what m is? Describe the process in details.

Let's assume Bob is author of the document, Alice is signer.

1. Bob computes a blinding factor, by calculating parameter *r*, such that $r \in Z_n$ with gcd(r, n) = 1. Then Bob takes Alice's public key ***e*** and encrypts *r* to obtain blinding factor:
   blinding factor = $r^e \ mod \ n$
2. Bob submits to Alice the product: $m \times r^e \ mod \ n$
3. Alice signs and Bob receives: $(m \times r^e)^d = \ m^d \times r^{ed} \ = m^d \ \times r \ mod \ n$
4. Bob removes the blinding factor by multiplying with the inverse of *r* from the right and obtains Alice's signature: $m^d \ = \ m^d \ \times r \ \times r^{-1} \ mod \ n$