**ICS 180: Introduction to Cryptography** 4/15/2004

# Homework 2

*Due Thursday, 4/22/2004 at the beginning of class.*

## 1 Security Definitions [10+20 points]

Definition of some security property often goes like this: We call some communication scheme $\Sigma$ *secure in the sense of resistance against attack of type "X"* if for all probabilistic polynomial time algorithms $A$, the probability that $A$ succeeds in an *"attack of type X"* against $\Sigma$ is negligibly small, i.e. it's a negligible function of the security parameter $\tau$.

For example, the definition of *one-way secure* encryption scheme $\Sigma = (KGen, Enc, Dec)$ has exactly this form, where *"attack of type X"* of $A$ against $\Sigma$ is the "decryption attack", defined as follows: (1) $KGen$ is executed on the security parameter $\tau$ to create key $k$, (2) random $m$ is picked in the messages space $\mathcal{M}$, (3) ciphertext $c$ is computed as $Enc(k, m)$, and finally (4) $A$ runs on input $c$ and outputs some string $m'$. We say that $A$ succeeds in this attack if $m' = m$.

### 1.1 [10 points]

Show a (trivial) PPT algorithm which succeeds with a non-zero but negligible probability in an attack against the "one way security" property of the one-time pad encryption scheme defined for message space $\mathcal{M} = \{0,1\}^\tau$ and key space $\mathcal{K} = \{0,1\}^\tau$, where $\tau$ is the security parameter.

Note that this means that even if a scheme is *perfectly secure*, let alone *one-way secure*, there nevertheless usually exist efficient attacks against it which succeed with *negligible* probability. This, in part, is why we usually cannot ask that that the probability of successful break of our scheme be zero for all efficient algorithms.

### 1.2 [bonus 20 points]

Let's show that the definitions of this type are "robust" in the following sense: Assume that a scheme $\Sigma$ is secure against "attack of type X" in the above sense, but that there nevertheless exists an efficient algorithm $A$ which *does* succeed in this attack but only with a negligible probability, for example $2^{-p(\tau)}$ for some polynomial $p(\cdot)$.

Consider a new efficient attack algorithm $A'$, which simply runs attack $A$ for some polynomial number of times, say $p'(\tau)$, and succeeds if *any* of these runs of $A$ return a succesful output. Argue why $A'$ is an *efficient* algorithm, and show that such polynomial-number of repetitions of attack against $\Sigma$ still has only negligible probability of success.

*[[Hint: First of all, your goal is to argue that the probability that $A'$ succeeds is smaller than some negligible function* for all large enough $\tau$*, i.e. for all $\tau$ larger than some $\tau_0$. Therefore all intermediate steps you make do not have to hold for all $\tau$'s, but only for all sufficiently large $\tau$'s. A convenient way to do this is to look at the probability that $A'$* fails*, and try to show that for all large enough $\tau$'s, this probability is larger than $1 - \epsilon$, where $\epsilon$ is some conveniently chosen negligible function.*

*You might use the following facts: (1)* $(1-\frac{1}{a_n})^{a_n} \approx \frac{1}{e}$ *for any* $a_n \to \infty$*, where* $e = 2.718...$*, (2)* $\frac{1}{4} < \frac{1}{e} < \frac{1}{2}$*, (3)* $(x^y)^z = x^{yz}$ *for all* $x, y, z$*, and therefore also* $x^z = (x^y)^{(z/y)}$ *for all* $x, y, z$*, (4) for all* $x, y, c > 0$ *inequality* $x > y$ *holds if and only if* $x^c > y^c$*, (5) for any polynomials* $g(\cdot), g'(\cdot)$*, inequality* $2^{g(\tau)} > g'(\tau)$ *holds for all large enough* $\tau$*'s, (6) therefore, in particular, for any polynomial* $g(\cdot)$*, function* $1/2^{g(\tau)}$ *is negligible.]]*

## 2 One-way security vs. indstinguishability [20 points]

Recall the definitions of "one way security" and "indistinguishability" of an encryption scheme. Recall the proof we did in the lecture which showed that if one-way secure encryption schemes exist at all, then there can be an encryption scheme which is one-way secure but not indistinguishable.

Now prove that if an encryption scheme is secure in the sense of indistinguishability then it is also secure in the sense of one-wayness for message space $\mathcal{M} = \{0,1\}^\tau$. This will show that indistinguishability is a strictly stronger security property of encryption than one-wayness.

You can prove this by proving the counterpositive, i.e. assume that some encryption scheme $\Sigma$ is *not* one-way secure, and then show that it is also *not* indistinguishable. If you assume that $\Sigma$ is not one-way secure then there exists a PPT algorithm $A$ that breaks one-wayness of $\Sigma$. Using such $A$ construct a PPT algorithm $A'$ s.t.

$$Prob[A'(m_0, m_1, c) = 1 \mid k \leftarrow KGen(1^\tau), (m_0, m_1) \leftarrow \{0,1\}^\tau, c \leftarrow Enc(k, m_1)]$$

is non negligible, while

$$Prob[A'(m_0, m_1, c) = 1 \mid k \leftarrow KGen(1^\tau), (m_0, m_1) \leftarrow \{0,1\}^\tau, c \leftarrow Enc(k, m_0)]$$

is negligible.

Argue that the existence of $A'$ leads to breaking the indistinguishability property of $\Sigma$.

## 3 Inverting ElGamal cryptosystem [25 points]

Consider (a variant of) an ElGamal cryptosystem, where $SK = (p, g, y, x)$ where $p$ is a large prime of size polynomial in the security parameter $\tau$ (similarly to the RSA modulus, 1024-bit $p$ is believed to offer about $2^{80}$ security, and therefore is often used today), $g$ is a *generator* of group $\mathbb{Z}_p^*$, $x$ is a random number in $\mathbb{Z}_{p-1}$, and $y = g^x \bmod p$. (See the handout on arithmetic modulo primes.)[1] The public key used for encryption is $PK = (p, q, y)$ and $SK = (p, q, y, x)$ is used to decrypt.

The encryption works as follows. The input message is broken-down into blocks $m$ s.t. each $m \in \mathbb{Z}_p^*$ and then each $m$ is encrypted individually as follows:

- $Enc(PK, m) = (c_1, c_2) = (g^r \bmod p,\ y^r * m \bmod p)$, where $r$ is a random number in $\mathbb{Z}_{p-1}$ picked by the encryption algorithm. (Each ciphertext is a pair of $(c_1, c_2) \in (\mathbb{Z}_p^*, \mathbb{Z}_p^*)$.)

[1] In particular, recall that $\mathbb{Z}_p^* = \{1, ..., p-1\}$, $\mathbb{Z}_{p-1} = \{0, ..., p-2\}$, and that if $g$ is a generator than for every element $y \in \mathbb{Z}_p^*$ there is a unique element $x \in \mathbb{Z}_{p-1}$ s.t. $y = g^x \bmod$. Therefore, in particular, if you pick $x$ at random in $\mathbb{Z}_{p-1}$, element $y = g^x \bmod p$ is itself random, i.e. uniformly distributed, in $\mathbb{Z}_p^*$.

- $m = Dec(SK, (c_1, c_2)) = c_2/(c_1)^x \bmod p$. (Check that this comes out right!)[2]

Assume that someone creates an (efficient) algorithm $A$ which decrypts ElGamal ciphertexts knowing just the public key, but only if $c_1$ starts with at least 5 leading zeroes, i.e. $c_1 = 00000.....$

What's the advantage of $A$ in breaking the one-wayness of ElGamal? Is it negligible?

It seems that one could counteract such an attack by modifying ElGamal encryption so that $c_1$ never starts with 00000 substring. The encryption procedure could simply pick another $r$ if $c_1$ happens to start with 00000.

However, show how that algorithm $A$ can be used as a black box to construct an algorithm $A'$ which (efficiently) decrypts *every* ciphertext, and not just those s.t. $c_1$ starts with 5 zeroes. What's the running time of $A'$ compared to the running time of $A$?

[2]Note: The security of ElGamal rests on the difficulty of computing discrete logarithms, because otherwise someone could compute $x = DL_g(y)$ and decrypt on their own. But in fact ElGamal rests on a stronger assumption of its own, namely that given $(p, g)$ and random $c_1, y \in \mathbb{Z}_p^*$, it is hard to compute $c_1^x \bmod p$ without knowing $x$ s.t. $y = g^x \bmod p$, i.e. without knowing $x = DL_g(y)$. This is similar to the case of RSA, which definitely needs the difficulty of factoring to be secure, but really requires a stronger assumption of its own.