Foundations of Cryptography

Notes of lecture No. 5*B* (given on Apr. 2nd)

Notes taken by Eyal Kushilevitz

Summary

In this (half) lecture we introduce and define the concept of *secure encryption*. (It is assumed that the reader is familiar with the notions of private-key and public-key encryption, but we will define these notions too for the sake of self-containment).

1. Secure encryption - motivation.

Secure encryption is one of the fundamental problems in the field of cryptography. One of the important results of this field in the last years, is the success to give formal definitions to intuitive concepts (such as secure encryption), and the ability to suggest efficient implementations to such systems. This ability looks more impressive when we take into account the strong requirements of these definitions.

Loosely speaking, an encryption system is called "secure", if seeing the encrypted message does not give any partial information about the message, that is not known beforehand. We describe here the properties that we would like the definition of encryption system to capture:

- **Computational hardness -** As usual, the definition should deal with efficient procedures. That is, by saying that the encryption does not give any partial information about the message, we mean that no efficient algorithm is able to gain such an information Clearly, we also require that the encryption and decryption procedures will be efficient.
- **Security with respect to any probability distribution of messages -** The encryption system should be secure independently of the probability distribution of the messages we encrypt. For example: we do not want a system which is secure with respect to messages taken from a uniform distribution, but not secure with respect to messages written in English. Namely, we do not want that a user of an encryption scheme will have to predetermine what is the probability distribution of messages for which he intend to use the system. Therefore, the designer of an encryption system has no idea about the probability distribution of messages that the user will send. (One possible idea, is to condense the messages space into a space with a uniform distribution. However, this may be a solution to the problem of how to construct encryption systems but can not be considered as a definition of such a system.)
- **Hardness of gaining any partial information -** It should be hard not only to find m given $E(m)$ but also to find *any* partial information about m, such as part of m bits etc. This requirement may seems to

be somewhat as a "paranoia", but it is needed since we would like to protect any "important" information. However, we do not know how the user intends to use the system and therefore we cannot define what is the "important" information. In particular, he may use the system in a way that the most "secret" part of his messages is exactly the partial information which is easy to find. Therefore we call an encryption system secure, only if any partial information is hard to find.

- **Hardness against a-priori information -** We require that even if the "enemy" has some a-priori information about the message (e.g. in what language it is written, or that the message is either "yes" or "no") then it will not help him to achieve significant information about the message, which is not follows from his a-priori knowledge. We remark that this requirement is very similar to the requirement of being robust against any probability distribution. We discuss this similarity later.

At this point of the course, since we assume that one-way functions do exist, one may suggest that we will use one-way functions as an encryption. We remark here that one-way functions may not satisfy the requirements stated above. Namely, if f is a one-way function it is only guaranteed that given $f(m)$ it is hard on the average to find m, assuming that m is taken from the uniform probability distribution. It is also not guaranteed that it is hard to gain some partial information about m, or that if we a-priori know certain bits of m we will not be able to find all the other bits. For example, the *RSA* function which is considered one-way, has the property that the Jacobi-symbol of the encrypted message is equal to that of the original message. Thus, if the Jacobi-symbol of m is an important information about the messages, it is not hidden by $RSA(m)$.

One should pay attention that if we have only a "small" number of possible messages, we can not expect a deterministic public-key encryption-system to be secure. This is because the enemy, given $E(m)$, may encrypt by himself all the possible messages and check which of them gives $E(m)$.

2. Formal definitions

We start with the formal definition of a public-key (private-key) encryption system. Next, we will define the notion of semantic security.

Definition 1: A *Public-key Encryption system* consists of three probabilistic polynomial-time algorithms (G,E,D) as follows:

1) G is an algorithm for generating keys. That is, $G(1^n)=(e,d)$ where e is the public-key, d is the private-key, n is a security parameter, and $|e|=|d|=n$.

2) E is an encryption algorithm and D is a decryption algorithm. For every message m of size $|m|=n$, and every pair (e,d) generated by G on input 1^n, and all the possible coin tosses of E,

$$D(E(m,e),d)=m \qquad (*)$$

The definition of *private-key encryption system* is the same except for requiring that $d=e$. We also remark that one can make the definition more liberal by requiring that (*) will be satisfied only for almost all coin tosses of the algorithms G,E and D.

We now turn to the definition of semantic-security (in the next lecture another, equivalent, notion of security will be defined). We start with an informal definition:

> An encryption system will be called *semanticly secure* if for every probability distribution of messages, everything that can be **efficiently computed** given the encrypted message, can be **efficiently computed** without it.

The main difference between this definition and Shannon's definition of security is that here we require security with respect to efficient computations while Shannon defined security from the point of view of information theory.

For giving the formal definition of security we also need a definition of samplable probability distributions.

Definition 2: A probability distribution $\{\pi_n\}_n$ is called *polynomial-time samplable* if there exists a probabilistic polynomial-time algorithm A such that for every $s\in\{0,1\}^n$

$$Pr\left[A(1^n)=s\right]=\pi_n(s)$$

We now give the definition of semantic security in both non-uniform and uniform formulation.

Definition 3: An encryption system (G,E,D) is called *semanticly secure* if for every probability distribution, π_n, every a-priori information function, h, every semantic function, f, and every probabilistic polynomial-time (non-uniform) algorithm, A, there exists a probabilistic polynomial-time (non-uniform) algorithm, A', such that for every $c>0$, and sufficiently large n (the size of messages),

$$Pr\left[A(E_e(m),e,h(m),1^n)=f(m)\right] < Pr\left[A'(h(m),1^n)=f(m)\right]+\frac{1}{n^c}$$

where the left probability is taken over the coin tosses of the algorithms A,E and G (e is the output of $G(1^n)$) and the probability distribution of messages (π_n), and the right probability is taken over the coin tosses of A' and the probability distribution of messages (π_n).

Definition 3': An encryption system (G,E,D) is called *uniformly semanticly secure* if for every samplable probability distribution, π_n, every polynomial-time computable a-priori information function, h, every semantic function, f, and every probabilistic polynomial-time algorithm, A, there exists a probabilistic polynomial-time algorithm, A', such that for every $c>0$, and sufficiently large n (the size of messages),

$$Pr\left[A(E_e(m),e,h(m),1^n)=f(m)\right] < Pr\left[A'(h(m),1^n)=f(m)\right]+\frac{1}{n^c}$$

where the left probability is taken over the coin tosses of the algorithms A,E and G (e is the output of $G(1^n)$) and the probability distribution of messages (π_n), and the right probability is taken over the coin tosses of A' and the probability distribution of messages (π_n).

We give here some remarks with respect to this definition:

- When dealing with a private-key encryption system, A does not get e as an input.
- If we take f as the identity function ($f(m)=m$) the definition requires that one, seeing $E(m)$, can not guess m significantly better than if he does not see $E(m)$.
- The definition has a meaning even when the semantic function f is not computable. This is because the algorithms A and A' do not try to compute f but only to approximate it. This means that $E(m)$ does not help to achieve approximation of $f(m)$ which is better than what can be achieved without using $E(m)$.
- Giving 1^n as an input to A', is needed in order to enable the (uniform) algorithm to run polynomial-time in the length of m (otherwise it may not be able even to write its output), and so that the length of m is always given as an a-priori information on m. It can be shown that no "efficient encryption scheme" can hide the length of messages encrypted by it.
- For the non-uniform definition, the a-priori information function h is not needed. Namely, the definition is equivalent even when A and A' do not get $h(m)$ as an input. This is since the a-priori information h can be combined with the probability distribution π_n to create a new probability distribution π'_n. The encryption system is assumed to be robust also with respect to this new probability distribution and this implies its security with respect to π_n and h. The exact construction of π'_n from π_n and h follows from the proof of a theorem, that will be given in the next lecture, claiming the equivalence of the "semanticly secure" notion to another notion of security called "indistinguishability security".
- Finally, we remark that in the non-uniform case we can consider only deterministic algorithms. This follows from the fact that if there is a probabilistic algorithm which finds $f(m)$ with a good probability, then by an averaging argument there exists a sequence of random coins on which this algorithm has a good probability of success. This sequence can be wired into the circuit.