0368.4162: Introduction to Cryptography Ran Canetti

Lecture 9 - Symmetric Encryption

5 February 2010 Fall 2010

Original Notes (2008) by: R. Levi, M. Rosen
Updated (2010) by: A. Kovelman

1 Introduction

Encryption, or guaranteeing secrecy of information, is perhaps the most basic task in cryptography, yet, it is the hardest to capture mathematically. We first deal with *symmetric encryption*, where the parties have a joint secret key. Next lecture will discuss *public key encryption*, but most of the basic principles and notions will be the same or very similar.

A pair of algorithms *(ENC, DEC)* is a symmetric encryption scheme if it guarantees correct decryption (validity): For any m in message domain *M, DEC (k, ENC (k, m)) = m* (except perhaps for negligible probability in say |k|, over the choice of k and the random choices of DEC and ENC). Next, we will want to define "secrecy".

How to define secrecy? It is easier to capture "lack of secrecy", i.e. what must not happen:

- Adversary should not learn the secret key.
- Adversary should not learn the message.
- Adversary should not learn parts of the message (e.g. the first bit of the message).
- Adversary should not learn any predicate of the message.
- Adversary should not be able to recognize any relations between encrypted messages (e.g. he shouldn't know if a message was sent twice).
- Adversary should not be able to do the above even if she has some prior knowledge on the message (e.g. she knows that the message is written in English or that it is either "buy" or "sell").

Notice that already from these requirements we can conclude that encryption needs to be either stateful or probabilistic; otherwise repeated message will have the same cipher texts.

Since the list is too long and we never know when we "get it all" we try a different approach: describe an ideal scheme, where security will be obviously guaranteed. We say that an encryption scheme is secure if it "behaves like the ideal scheme" in the eyes of any feasible observer. Let's formalize this definitional approach.

2 Semantic Security

The statement that a symmetric encryption scheme is secure can be made by saying that an encryption scheme is indistinguishable from some ideal scheme (which we know to be secure). Our ideal scheme will be such that the adversary can't possibly learn anything about the encrypted message by observing the cipher text, except some amount of data that we leak intentionally (this leaked data will be called a *leakage function*; a typical example of a leakage function is the size of the message); we will then say that the encryption scheme is secure with respect to the specific leakage function.

We formalize the definition in three steps:

1. Formalize the real-life setting we're addressing
2. Formalize the ideal scheme
3. Formalize the notion of "behaves like the ideal scheme"

2.1 Real Life Scheme

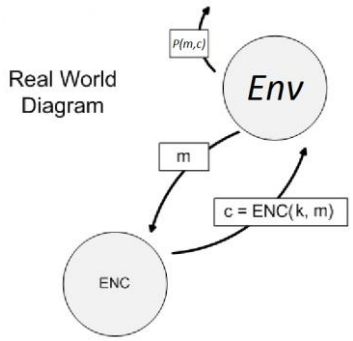

Figure 1: Real life scheme

Assume a (hostile) environment Env that generates messages based on some distribution D and attempts to learn something based on their corresponding cipher text. We can model the encryption process as follows:

1. Env draws *m* from distribution *D* and sends it to *ENC*
2. Env receives $c = ENC(k, m)$
3. Env computes $P(m, c)$, where *P* is a predicate (intuitively, $P(m, c) = 1$ represents that the information learned from *c* by Env on *c*'s plaintext agrees with *m*)

[See Figure 1]

2.2 Ideal Scheme

The ideal encryption process is the same as the real life process, except that Env doesn't receive the cipher *c*. Instead, the output of some leakage function $l(m)$ of the plaintext is given to a simulator S, which then outputs c' based solely on $l(m)$. $l(m)$ represents the information that the scheme is allowed to leak on a plaintext (a common example is *l(m)* = |*m*|). More precisely:

1. Env sends m to *ENC*
2. Some information $l(m)$ leaks from *ENC* to the simulator *S*
3. *S* computes c' from $l(m)$ and sends c' to Env
4. Env computes $P(m, c')$, where *P* is a predicate

[See Figure 2]

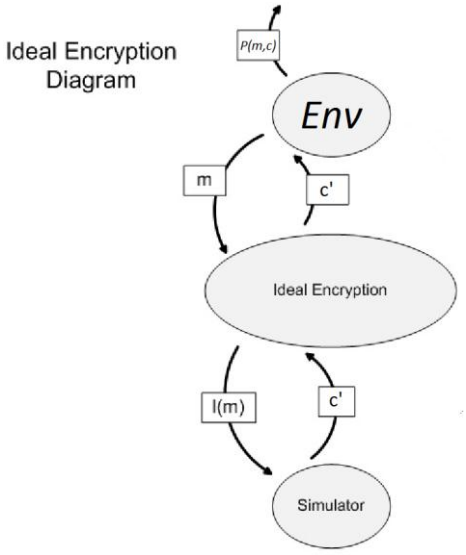

Figure 2: Ideal scheme

2.3 "Behaves Like the Ideal Scheme"

Intuitively, when saying that a scheme "behaves like the ideal scheme", we want to say that from the point of view of any external observer the real scheme looks like the ideal scheme. Namely, there is a simulator *S* that simulates the real scheme s.t. any environment Env that interacts with either the real scheme or with S cannot tell the difference between the two cases. Formally:

Definition 2.1. *A symmetric encryption scheme (ENC, DEC) is* ***semantically secure*** *with respect to message domain* $M = \{M_n\}_{n\in N}$ *and leakage function l, if there exists a polytime simulator S s.t. for any environment* Env *and for any polytime predicate P :*

$$\Big|Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c \leftarrow ENC(k,m), P(m,c) = 1] - Pr_{k\in\{0,1\}^n}\big[m \leftarrow Env(1^n), c' \leftarrow S\big(l(m)\big), P(m,c) = 1\big]\Big| < v(n)$$

where $v(n)$ *is some negligible function.*

Note the new notation: the experiment is described inside the probability brackets. The probability is taken also over the random choices of the relevant algorithms.

3 Security by Indistinguishability

Even though semantic security seems like very intuitive notion, it is certainly cumbersome to work with. So instead we will see an alternative definition which is considerably simpler, and ends up being equivalent to semantic security. The definition involves a simple game where the adversary is tested for the ability to guess which message is encrypted in a given cipher text. That is, given an encryption scheme *(ENC, DEC)* for domain *M*, leakage function *l* and an adversary *B*, define the following game:

1. *B* chooses two messages $m_0, m_1 \in M$ s.t. $l(m_0) = l(m_1)$
2. A bit $b \leftarrow \{0,1\}$ and a key $k \in \{0,1\}^n$ are chosen at random and $c = ENC(k, m_b)$ is computed
3. *B* receives c
4. *B* outputs a bit b'

We say that B wins the game if $b = b'$.

Definition 3.1. ***Security by Indistinguishability***
A symmetric encryption scheme (ENC, DEC) *is secure by indistinguishability with respect to message domain* $M = \{M_n\}_{n\in N}$ *and leakage function* l *if for any polytime adversary B we have:*

$$Pr[k \leftarrow \{0,1\}^n, (m_0, m_1, s) \leftarrow B(1^n), b \leftarrow \{0,1\}, b' \leftarrow B(1^n, ENC(k, m_b), s) \mid l(m_0) = l(m_1) \cap b = b'] < \frac{1}{2} + v(n)$$

For some negligible function $v(n)$.

We add the variable s in the above definition to emphasize that B keeps an internal state throughout the whole experiment; here s represents the internal state of B. This definition indeed seems much simpler: no simulators and no predicates to learn.

For the following theorem, we assume that l is sampleable (i.e. given a value v in the range of l it is possible to find in polytime an $m \in M$ s.t. $l(m) = v$).

Theorem 3.2. *A symmetric encryption scheme* $E = (ENC, DEC)$ *is semantically secure with respect to domain* M *and sampleable leakage function* l *if and only if* E *is secure by indistinguishability for* M *and* l.

Proof. Let's denote security by indistinguishability IND and semantic security SEM.

IND ⇨ SEM
Intuition: To show that *E* is also *SEM*, we need to construct, for each environment Env that sees an encryption of a message m, a simulator S that generates some cipher test c', that satisfies any external predicate P with respect to the encrypted message - without seeing $c = ENC(k, m)$. The idea is that S can simply encrypt some arbitrary dummy message m'. Env will not be able to say something meaningfully different on $ENC(k, m')$ than on $ENC(k, m)$, or else Env can be used to win the indistinguishability game for E.

Let $E = (ENC, DEC)$ be secure by indistinguishability with respect to message domain M and sampleable leakage function l.
So, by definition, for any polytime adversary B:

$$Pr[k \leftarrow \{0,1\}^n, (m_0, m_1, s) \leftarrow B(1^n), b \leftarrow \{0,1\}, b' \leftarrow B(1^n, ENC(k, m_b), s) \mid l(m_0) = l(m_1) \cap b = b'] < \frac{1}{2} + v(n)$$

For some negligible function $v(n)$.

We build S as follows:
Given input $1^n, l$:
1. S finds m' s.t $l(m') = l$
2. S chooses $k \leftarrow \{0,1\}^n$
3. S outputs $c' = ENC(k, m')$

Assume by contradiction that E is not semantically secure with respect to domain M and l. Then there exists an Env s.t. for any S (also the S we built):

$$|Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c \leftarrow ENC(k, m), v = Env(c), P(m, v) = 1] \quad (1)$$
$$-Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c' \leftarrow S(l(m)), v = Env(c'), P(m, v) = 1]| > \varepsilon \quad (2)$$

We'll get our contradiction by constructing an adversary B that wins the distinguishing game with probability $> \frac{1}{2} + \frac{\varepsilon}{2}$:
1. B chooses at random $r \in \{0,1\}$

2. B chooses two messages m_0, m_1 s.t. $m_r \leftarrow Env(1^n)$, m_{1-r} is the message chosen by S
3. B receives $c = ENC(k, m_b)$ for $b \in \{0,1\}$, $k \in \{0,1\}^n$
4. B computes $v = Env(c)$; if $P(m, v) = 1$ then *B* outputs $b' = r$, else *B* outputs $b' = 1 - r$

The idea here is that if *c* is an encryption of m_r then we are in case (1). If *c* is an encryption of m_{1-r} then we are in case (2). More precisely:

$$Pr[B\ wins\ the\ distinguishing\ game] = \quad (3)$$
$$= Pr[B\ wins | b = r] * Pr[b = r] + Pr[B\ wins | b = 1 - r] * \Pr[b = 1 - r] \quad (4)$$
$$= \frac{1}{2}(Pr[B\ wins | b = r] + Pr[B\ wins | b = 1 - r]) \quad (5)$$
$$= \frac{1}{2}(\Pr[B\ wins | b = r] + 1 - \Pr[B\ loses | b = 1 - r]) \quad (6)$$
$$= \frac{1}{2}(Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c \leftarrow ENC(k, m_r), v \leftarrow Env(c), P(m, v) = 1] \quad (7)$$
$$- Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c' \leftarrow ENC(k, m_{1-r}), v \leftarrow Env(c'), P(m, v) = 1]) \quad (8)$$
$$+ \frac{1}{2} \quad (9)$$
$$= \frac{1}{2}(\ (1) + 1 - (2)\ ) > \frac{1}{2}(1 + \varepsilon) = \frac{1}{2} + \varepsilon \quad (10)$$

(6) is correct since $Pr[B\ wins | b = 1 - r] + Pr[B\ loses\ | b = 1 - r] = 1$.
(7) is correct because if *B* wins when $b = r$, then *B* outputs $b' = r$ and $P(m_r, v) = 1$.
(8) is correct because if *B* loses when $b = 1 - r$, then *B* outputs $b' = r$ (he lost because $b = r \neq b'$) and $P(m_r, v) = 1$.
We got a contradiction for E being IND.

SEM ⇨ IND

We show that the ability to win the game can be translated to the ability to predict a non-trivial predicate of the message, under a certain distribution. That is, we show: $\neg IND \Rightarrow \neg SEM$.

Assume *E* is not secure by IND. Therefore we have an adversary *B* that wins the distinguishing game with probability $> \varepsilon + \frac{1}{2}$. We want to show that *E* is not secure by SEM, therefore we need to construct $Env,\ P$ such that no simulator S can satisfy (1), (2). (Note that this is slightly stronger than what we need - since Env and *P* won't depend on S). Let m_0, m_1 be the messages generated by *B*. Note that m_0, m_1 are fixed since *B* is w.l.o.g. deterministic.

1. Env obtains random $b \leftarrow \{0,1\}$, $(m_0, m_1, s) \leftarrow B(1^n)$, outputs m_b and receives cipher text *c*
2. B outputs $b' = B(s, c)$ and passes b' to Env
3. Env outputs 1 iff $b' = b$

By assumption on B, Env outputs 1 with probability $> \varepsilon + \frac{1}{2}$. However, the view of *S* is completely independent of b, thus it can make output 1 with probability at most $\frac{1}{2}$ Therefore we get, for any simulator *S*:

$$\begin{aligned} &Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c \leftarrow ENC(k, m), v \leftarrow Env(c), P(m, v) = 1] \\ &- Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c \leftarrow S(l(m)), v \leftarrow Env2(c), P(m, v) = 1] \\ &= Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c \leftarrow ENC(k, m), b' \leftarrow B(c), b' = b] \\ &- Pr_{k\in\{0,1\}^n}[m \leftarrow Env(1^n), c \leftarrow S(l(m)), v \leftarrow Env(c), P(m, v) = 1] \\ &> \varepsilon + \frac{1}{2} - \frac{1}{2} = \varepsilon \end{aligned}$$

Therefore, E is not secure by SEM. □

4 Security for Multiple Encryptions, CPA security

The above definitions consider only the case where the adversary sees a single encryption of some message. When the adversary sees multiple messages it makes sense to ask how the messages to be encrypted are chosen. Two options are:

- Non-adaptively: all messages chosen in advance, before any cipher text is known
- Fully adaptively: each message is chosen after seeing the previous cipher text

For simplicity, we'll formulate IND-style definitions for these cases first. For the non-adaptive case, we modify the distinguishing game as follows:

Given an encryption scheme for domain M and leakage function l and an adversary B, define the following IND-KPA game (for indistinguishability under known plaintext attack):

1. *B* chooses *p* pairs of messages $m_{1,0}, m_{1,1}, \dots, m_{p,0}, m_{p,1} \in M$ s.t. for all *i*: $l(m_{i,0}) = l(m_{i,1})$
2. *B* receives $ENC(k, m_{1,b}), \dots, ENC(k, m_{p,b})$ for a random bit b
3. *B* outputs bit b'

B is said to win the game if $b = b'$.

For the adaptive case, the IND-CPA game (for indistinguishability under chosen plaintext attack) is the same, except that *B* generates the next pair of message only after seeing the previous cipher text.

Definition 4.1. *Let E = (ENC, DEC) be a symmetric encryption scheme. E is called* ***IND-KPA secure*** *(resp.,* ***IND-CPA secure****) if any polytime adversary wins the IND-KPA (resp., IND-CPA) game with probability at most* $\frac{1}{2} + v(n)$ *for some negligible function* $v(n)$*.*

Now, define the real and ideal schemes for **SEM-KPA security**: the basic idea is that we allow the adversary to output a set of messages $\{m_i\}$ and receive a set of matching cipher texts $\{c_i\}$, with differences between the real and ideal scheme just as in the definition of SEM security for a single message.

The real scheme proceeds as follows: let Env be an environment that generates messages based on some distribution D and attempts to learn something based on their corresponding cipher text. We can model the encryption process as follows:

1. Env draws $\{m_i\} = \{m_1, m_2 \dots m_t\}$ from distribution *D* and sends it to *ENC*
2. Env receives $\{c_i\} = \{ENC(k, m_1), ENC(k, m_2) \dots ENC(k, m_t)\}$
3. Env computes $P(\{m_i\}, \{c_i\})$, where *P* is a predicate (intuitively, $P(\{m_i\}, \{c_i\}) = 1$ represents that the information learned from $\{c_i\}$ by Env on *c*'s plaintext agrees with $\{m_i\}$)

Ideal scheme
Similarly to the SEM definition of the ideal scheme, the ideal encryption passes the output of the leakage function $l(m)$ to the simulator *S*, which then fabricates the cipher text; this occurs for each message in the set $\{m_i\}$. More precisely:

1. Env sends $\{m_i\}$ to *ENC*
2. Some information $\{l(m_i)\}$ leaks from *ENC* to the simulator *S*
3. *S* computes $\{c'_i\}$ from $\{l(m_i)\}$ and sends $\{c'_i\}$ to Env
4. Env computes $P(\{m_i\}, \{c_i\})$, where *P* is a predicate

Definition 4.2. *A symmetric encryption scheme (ENC, DEC) is* ***SEM-KPA secure*** *with respect to message domain* $M = \{M_n\}_{n \in N}$ *and leakage function l, if there exists a polytime simulator S s.t. for any environment* Env *and for any polytime predicate P :*

$$\left|Pr_{k\in\{0,1\}^n}[\{m_i\} \leftarrow Env(1^n), \{c_i\} = \{ENC(k, m_i)\}, P(\{m_i\}, \{c_i\}) = 1]\right.$$
$$\left. - Pr_{k\in\{0,1\}^n}[\{m_i\} \leftarrow Env(1^n), \{c'_i\} \leftarrow S(\{l(m_i)\}), P(\{m_i\}, \{c_i\}) = 1]\right| < v(n)$$

where $v(n)$ is some negligible function.

The definition of **SEM-CPA security** is similar to that of SEM, except that we allow the adversary to repeat steps (1)-(2) (output message and receive its cipher text) as much as they demand. Formally:

Real scheme:
1. Env draws *m* from distribution *D* and sends it to *ENC*
2. Env receives $c = ENC(k, m)$
3. Env may return to step (1) or continue to the next step
4. Env computes $P(T, C)$, where *P* is a predicate, T is the ordered set of all plaintexts and C is the ordered set of all cipher texts received by Env (intuitively, $P(T, C) = 1$ represents that the information learned from *C* by Env on *C*'s plaintext agrees with *T*)

Ideal scheme:
1. Env sends m to *ENC*
2. Some information $l(m)$ leaks from *ENC* to the simulator *S*
3. *S* computes c' from $l(m)$ and sends c' to Env
4. Env may return to step (1) or continue
5. Env computes $P(T, C)$, where *P* is a predicate, T is the ordered set of all plaintexts and C is the ordered set of all cipher texts received by Env

Definition 4.3. *A symmetric encryption scheme (ENC, DEC) is **SEM-CPA secure** with respect to message domain $M = \{M_n\}_{n\in N}$ and leakage function l, if there exists a polytime simulator S s.t. for any environment Env and for any polytime predicate P :*

$$\left|Pr_{k\in\{0,1\}^n}[P(T, C) = 1 \mid T = \{t_i \leftarrow Env(1^n, s)\}, C = \{c_i \leftarrow ENC(k, t_i)\}]\right.$$
$$\left. - Pr_{k\in\{0,1\}^n}[P(T, C') = 1 \mid T = \{t_i \leftarrow Env(1^n, s)\}, C' = \{c'_i \leftarrow S(k, l(t_i))\}]\right|$$
$$< v(n)$$

where $v(n)$ is some negligible function.

Note the abuse of notation: $l(T)$ stands for $\{l(t_i)|t_i \in T\}$, etc. Also note that Env receives the argument s that stands for the state of the environment as it changes throughout the game.

Fact 4.4. IND security does not imply IND-KPA security.
Fact 4.5. Neither does IND-KPA security imply IND-CPA security.
Proof left as Homework.

5 Protection against Active Adversaries and Secure Channels

So far we discussed security against an adversary that only listens to the communication. What about an adversary that can also inject or modify messages, as in the case of MACs?

There are many notions in the literature of secrecy against active attacks. Their applicability to practice is not always clear in the case of private-key encryption, so we'll defer treating these notions to the case of public-key encryption, where they are better motivated. Instead, we'll concentrate on the concrete task of realizing secure communication channels. Here we have two concerns:

- Maintaining authenticity of messages
- Maintaining secrecy of messages

In order to formalize the notion of secure-channels scheme we can either extend the game of IND-CPA or extend the semantic security approach in a way that captures adversary's capability to modify messages. Since the semantic approach formulation is more convincing and will be useful for us in the future we'll extend the real life and ideal settings to capture the adversary's new capability.

A secure-channels scheme consists of two algorithms SEND and REC. We'll proceed to formalize this definition in three steps, as we did before:

5.1 Real Life Scheme

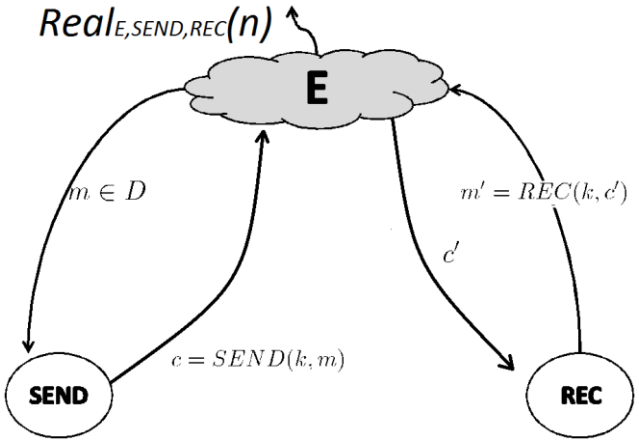

Figure 3: Real secure connection

The real world interaction proceeds as follows:

1. A key $k \in \{0,1\}^n$ is chosen and given to *SEND* and *REC*
2. *E* sends some value m to *SEND* and receives $c = SEND(k, m)$
3. *E* sends some value c' to *REC* and receives $m' = REC(k, c')$
    a. Notice that REC is allowed to return an error code instead of m'

E repeats this process until at some point it outputs a value. We'll see that w.l.o.g. this value can be one bit. (Intuitively, this bit says whether *E* thinks it is in the Real or Ideal interaction). See Figure 3. Let $Real_{E,SEND,REC}(n)$ denote the output of *E* in this game; $Real_{\mathrm{E,SEND,REC}}$ denotes the ensemble $\{Real_{E,SEND,REC}(n)\}_{n \in N}$.

5.2 Ideal Scheme

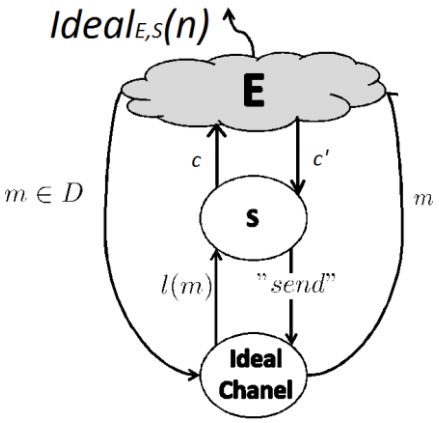

Figure 4: Ideal secure connection

In the ideal setting the environment *E* communicates with both the ideal secure channel and a simulator S (as described below). Similarly to the case of the symmetric encryption, S receives only the leakage function $l(m)$ of the original message, and creates some cipher text based on it, thus assuring secrecy. The ideal scheme proceeds as follows:

1. *E* sends some plaintext *m* to the ideal channel
    a. S receives $l(m)$ and passes some string *c* to *E*
2. *E* sends some cipher text c' to S
    a. S receives c' and either returns an error or sends a "Send" command to the ideal channel (which causes the channel to pass *m* to *E*)

E repeats this process until at some point it outputs a value. Notice that the simulator *S* is allowed to send an error code instead of "send" to emulate the case that *REC* returned an error code in the real-life scheme. See Figure 4.
Let $Ideal_E(n)$ denote the output of *E* in this game; $Ideal_E$ denotes $\{Ideal_E(n)\}_{n\in N}$.

5.3 "Behaves Like the Ideal Scheme"

The definition is formulated as in the case of encryption:

Definition 6.1. *A pair of interactive algorithms (SEND, REC) is a secure channel protocol if for any environment E there exists a simulator S such that:*

$$Real_{E,SEND,REC} \approx_c Ideal_E$$

5.3 Alternative definition

As in the case of symmetric encryption, define an equivalent notion of security via a game: an environment *E* plays the following game against a pair of protocols *(SEND, REC)* and a leakage function $l(m)$:

1. A key $k \leftarrow \{0,1\}^n$ is chosen and given to *SEND,REC*
2. *E* may output a plaintext *m* and receive $c = SEND(k, m)$
3. *E* may output a cipher text *c'* and receive $m' = REC(k, c')$
4. *E* may repeat steps (2) and (3) as needed
5. *E* outputs a pair of plaintexts (m_0, m_1), receives $c = SEND(k, m_b)$ for random

(externally set) bit b

6. *E* may repeat steps (2) and (3) as needed
7. *E* outputs a bit b'

OR

8. *E* outputs a cipher text c'

E wins the secure channels game if:

$\Pr[\, k \leftarrow \{0,1\}^n, b' \leftarrow E(1^n), b' = b] > \frac{1}{2} + \varepsilon(n)$

for some non-negligible function $\varepsilon(n)$, or:

$\Pr[k \leftarrow \{0,1\}^n, c' \leftarrow E(1^n), \exists m\ s.t.\ c' = SEND(k,m) \cap E\ didn't\ invoke\ SEND(m)] > \varepsilon(n)$

for some non-negligible function $\varepsilon(n)$.

Theorem 6.1. *A secure channel scheme is secure according to the first definition (semantic security) if and only if it is secure according to the second.*