0368.4162: Introduction to Cryptography Ran Canetti

Lecture 9 - Symmetric Encryption

29 December 2008 Fall 2008

Scribes: R. Levi, M. Rosen

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 ciphertexts.
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

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

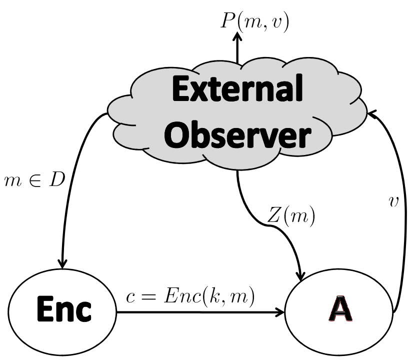

Figure 1: Real life scheme

Assume external observer E. We can model the encryption process as follows:

- E draws m from distribution D and sends it to ENC
- Some information $Z(m)$ leaks to an adversary A
- A receives $c = ENC(k, m)$
- A computes v from $c, Z(m)$. (Intuitively, v represents some value that A learned on m.)
- E computes $P(m, v)$, where P is a predicate. Intuitively, $P(m, v) = 1$ represents that the adversary A indeed learned something about m

See Figure 1

2.2 Ideal Scheme

The ideal encryption process is the same as the real life process, except that the adversary, now called S (simulator), doesn't receive the cipher c. Instead, it sees some leakage function l. This function represents the information that the scheme is allowed to leak on plaintext (a common example is $l(m) = |m|$). More precisely:

- E sends $m \leftarrow D$ to ENC
- Some information $Z(m), l(m)$ leaks to the adversary S
- S computes v from $l(m), Z(m)$, and sends v to E
- E computes $P(m, v)$, where P is a predicate

See Figure 2

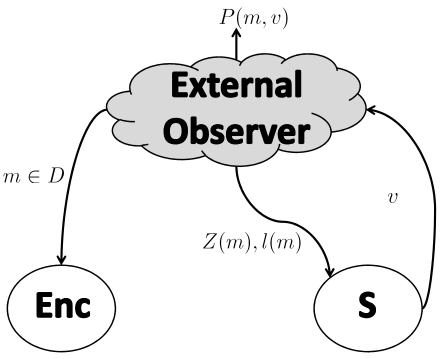

Figure 2: Ideal scheme

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, for any adversary A that interacts with the real scheme there could be an adversary S that interacts with the ideal scheme s.t no external process can 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 for any polytime adversary A there exists a polytime simulator S s.t for any sampling algorithm D s.t $support(D(1^n)) \subseteq M_n$, any polytime auxiliary information function Z and any polytime predicate P:*

$$\begin{aligned} &Pr_{k \in \{0,1\}^n}[m \leftarrow D(1^n), c \leftarrow ENC(k,m), v \leftarrow A(c, Z(m)), P(m,v) = 1] \\ &\quad - Pr[m \leftarrow D(1^n), v \leftarrow S(l(m), Z(m)), P(m,v) = 1] \quad < \quad \nu(n) \end{aligned}$$

where $\nu(n)$ 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.

Remark 2.2. *Since in this definition the scheme is secure for any auxiliary input functions Z, it doesn't matter if we modal A and S as uniform or non-uniform polytime adversaries (the advice can be supplied by Z).*

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 ciphertext. That is, given an encryption scheme (ENC, DEC) for domain M, leakage function l and an adversary B, define the following game:

- B chooses two messages $m_0, m_1 \in M$ s.t $l(m_0) = l(m_1)$
- 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
- B receives c
- 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:

$$Pr[k \leftarrow \{0,1\}^n, b \leftarrow \{0,1\}, (m_0, m_1, s) \leftarrow B(1^n), b' \leftarrow B(1^n, ENC(k, m_b), s) \textit{ s.t } l(m_0) = l(m_1) \textit{ and } b = b'] < \frac{1}{2} + \nu(n)$$

For some negligible function $\nu(n)$.

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 distributions over messages, no partial information functions, 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 $\Rightarrow$ SEM
Intuition: To show that E is also SEM, we need to construct, for each adversary A that sees an encryption of a message m, an adversary S ("simulator") that does "just as good as A" in generating a value 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 run A on an encryption of some arbitrary dummy message m', and output whatever A does. A will not be able to say something meaningfully different on $ENC(k, m')$ than on $ENC(k, m)$, or else A can be used to win the indistinguishability game for E.

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

$$Pr[k \leftarrow \{0,1\}^n, b \leftarrow \{0,1\}, (m_0, m_1, s) \leftarrow B(1^n), b' \leftarrow B(1^n, ENC(k, m_b), s) \text{ s.t } l(m_0) = l(m_1) \text{ and } b = b'] < \frac{1}{2} + \nu(n)$$

For some negligible function $\nu(n)$.
We build S by using A as follows:
Given input 1^n,l,z:

- S Finds m' s.t $l(m') = l$
- S Chooses $k' \in \{0,1\}^n$, computes $c' = ENC(k', m')$
- S Computes $v = A(c', z)$
- S Outputs v

Assume by contradiction that E is not semantically secure with respect to domain M and l. Then there are A, D,Z,P s.t for any S (also the S we built):
$(*)$

$$Pr_{k\in\{0,1\}^n,ENC}[m \leftarrow D(1^n), c \leftarrow ENC(k, m), v \leftarrow A(c, Z(m)), P(m, v) = 1] \tag{1}$$
$$- \quad Pr[m \leftarrow D(1^n), v \leftarrow S(l(m), Z(m)), P(m, v) = 1] > \epsilon. \tag{2}$$

We'll get our contradiction by constructing an adversary B that wins the distinguishing game with probability $> \frac{1}{2}+\frac{\epsilon}{2}$:

- B chooses two messages m_0, m_1 s.t $m_0 \leftarrow D$, m_1 is the message chosen by S
- B receives $c = ENC(k, m_b)$ for $b \in \{0,1\}$, $k \in \{0,1\}^n$
- B chooses at random $r \leftarrow \{0,1\}$ and computes $v = A(Z(m_r), c)$
- If $P(m_r, v) = 1$ 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:

$$
\begin{aligned}
&Pr[B \text{ wins the distinguishing game}] = && (3)\\
&= Pr[B \text{ wins}|b=r]\cdot Pr[b=r] + Pr[B \text{ wins}|b=1-r]\cdot Pr[b=1-r] && (4)\\
&= \frac{1}{2}\cdot(Pr[B \text{ wins}|b=r] + Pr[B \text{ wins}|b=1-r]) && (5)\\
&= \frac{1}{2}\cdot(Pr[B \text{ wins}|b=r] + 1 - Pr[B \text{ loses}|b=1-r]) && (6)\\
&= \frac{1}{2}\cdot(Pr_{k\in\{0,1\}^n,ENC}[m\leftarrow D(1^n), c\leftarrow ENC(k,m_r), v\leftarrow A(c,Z(m_r)), P(m,v)=1] && (7)\\
&\qquad - Pr_{k\in\{0,1\}^n,ENC}[m\leftarrow D(1^n), c'\leftarrow ENC(k,m_{1-r}), v\leftarrow A(c',Z(m_r)), P(m,v)=1]) && (8)\\
&\qquad\qquad + \frac{1}{2} && (9)\\
&= \frac{1}{2}({}^{(1)}(*) + 1 -{}^{(2)}(*)) > \frac{1}{2}(1+\epsilon) = \frac{1}{2}+\frac{\epsilon}{2}. && (10)
\end{aligned}
$$

(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 $b' = r \Longrightarrow P(m_r, v) = 1$.
(8) is correct because if B loses when $b = 1-r$, then B outputs $b' = r$ (he lost because $b = r \neq b'$), $\Longrightarrow P(m_r, v) = 1$.
We got a contradiction for E being IND.

SEM $\Rightarrow$ 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 > $\epsilon + \frac{1}{2}$. We want to show that E is not secure by SEM, therefore we need to construct: A, D, Z, P such that no simulator S can satisfy $(*)$. (Note that this is slightly stronger than what we need - since D, Z 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.

- Distribution D is:
  $Pr_{m\leftarrow D(1^n)}(m = m_0) = Pr_{m\leftarrow D(1^n)}(m = m_1) = \frac{1}{2}$ (D's support is $\{m_0, m_1\}$, with probability $\frac{1}{2}$ for each)
- Z is the null function
- Adversary A simply runs B and outputs the bit b' that B outputs
- $P(m_b, b')$ outputs 1 iff $b = b'$

By assumption on B, A makes P output 1 with probability $> \frac{1}{2} + \epsilon$. 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:

$$\begin{aligned}
&Pr_{k\in\{0,1\}^n,ENC}[m \leftarrow D(1^n), c \leftarrow ENC(k,m), v \leftarrow A(c, Z(m)), P(m,v) = 1] \\
&\quad - Pr_{k\in\{0,1\}^n,ENC}[m \leftarrow D(1^n), v \leftarrow S(l(m), Z(m)), P(m,v) = 1] \\
&= Pr_{k\in\{0,1\}^n,ENC}[m \leftarrow D(1^n), c \leftarrow ENC(k,m), b' \leftarrow B(c, \overbrace{0}^{Z(\cdot)=0})), b = b'] \\
&\quad - Pr_{k\in\{0,1\}^n,ENC}[m \leftarrow D(1^n), v \leftarrow S(l(m), 0), P(m,v) = 1] > \frac{1}{2} + \epsilon - \frac{1}{2} = \epsilon.
\end{aligned}$$

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 ciphertext is known
- Fully adaptively: each message is chosen after seeing the previous ciphertext

For simplicity, we'll formulate only IND-style definitions for these cases. SEM-style definitions can be constructed analogously. 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):

- B chooses p pairs of messages $m_{1,0}, m_{1,1}, ..., m_{p,0}, m_{p,1} \in M$ s.t for all i, $l(m_{i,0}) = l(m_{i,1})$
- B receives $ENC(k, m_{1,b}), ..., ENC(k, m_{p,b})$ for a random bit b
- B outputs a 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 ciphertext.

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 $1/2 + v(n)$ for some negligible function $v(n)$.*

Fact 4.2. *If E is IND secure and encryption is stateless, then it is IND-KPA secure.*

Fact 4.3. *IND-KPA security does not imply IND-CPA security (even for stateless schemes).*

Proof left as Homework.

5 Symmetric Encryption Schemes

Fact 5.1. *The One-Time-Pad scheme is IND-CPA secure.*

Explanation: Consider an adversary B that plays the IND-CPA game. Since the ciphertext generated by the One-Time-Pad scheme is distributed independently from the plaintext, B's advantage in the game is 0, regardless of B's computational power.
Instead of One-Time-Pads we use block ciphers. The various ways to use block ciphers in order to encrypt are called *modes of operation*. We'll see that some of these modes (but not all) are IND-CPA secure:

5.1 Counter Mode - CM

Let $F = \{F_n\}_{n \in N}, F_n = \{f_k\}_{k \in \{0,1\}^n}, f_k : \{0,1\}^{b(n)} -> \{0,1\}^{b(n)}$ be a PRF family ensemble. $b(n)$ is the block size, $M = \{0,1\}^*$ and $l(m) = |m|$. The scheme:

```
Initialize counter $t = 0$.
To encrypt a message $m = m_1....m_l(|m_i| = b)$ do:
    for $i = 1..l$
        $c_i = m_i \oplus f_k(t)$;
        $t = t + 1$
    output $c = c_1, ..., c_l$ (where $c_l$ is truncated to the length of $m_l$) save current value of $t$
```

Notice that the encryption is stateful.

Theorem 5.2. *CM is IND-CPA secure.*

Proof. Consider the case where f_k is a random function from $\{0,1\}^{b(n)} \rightarrow \{0,1\}^{b(n)}$. Since f_k is applied to each input only once the scheme is essentially the One-Time-Pad scheme. Assume there exists an adversary B that wins the IND-CPA game w.p. $1/2 + \epsilon$. Then we can distinguish between F_n and a random function with success probability of ϵ by emulating the scheme for B. More precisely, given a function g:

- Receive m_0, m_1 from B
- Pick $b \leftarrow \{0,1\}$ randomly and calculate c by encrypting m_b using the CM scheme with g as the random function
- Send c to B
- B outputs a bit b'
- Output 1 iff $b = b'$

□

5.2 Randomized Counter Mode - RCM

Operation similar to CM, except that the initial point of the counter is chosen at random for each new message.

```
To encrypt a message $m = m_1, ..., m_l(|m_i| = b)$ do:
    choose $t \leftarrow [1..2^b]$
    for $i = 1..l$
        $c_i = m_i \oplus f_k(t)$;
        $t = t + 1$
    output $c = t, c_1, ..., c_l$ (where $c_l$ is truncated to the length of $m_l$)
```

Theorem 5.3. *RCM is IND-CPA secure.*

Proof. Same principle as in CM, except emulating the scheme for B requires implementing the random choice of t. The proof works since the probability of collision is negligible. □

5.3 Electronic Codebook - ECB

```
To encrypt a message $m$ do:
    output $f_k(m)$
```

ECB is IND-CPA insecure: adversary chooses two pairs $(0,1)$ and $(0,2)$, if the two ciphertext are the same then $b = 0$.

5.4 Cipher Block Chaining Mode - CBC, stateful version

```
Choose initial value $IV = 0$
To encrypt a message $m = m_1....m_l$ $(|m_i| = b)$ do:
    $c_0 = IV$
    for $i = 1..l$
        $c_i = f_k(m_i \oplus c_{i-1});$
    $IV = c_l$
    output $c = c_1, ..., c_l$ keep $IV$ in state
```

CBC is IND-CPA insecure: Since the adversary knows the IV CBC is essentially reduced to ECB.
The adversary chooses two pairs: $(0, 0)$, gets ciphertext c, and then asks for $(c, 0)$. If B gets c again as ciphertext then the plaintext is 0 and B outputs $b' = 1$. Notice that in attacking the stateful CBC mode, the adversary uses the ability to choose messages adaptively.

5.5 CBC mode, stateless version

The operation similar to stateful version, except that c_0 is chosen at random for each message.

```
To encrypt a message $m = m_1....m_l$ $(|m_i| = b)$ do:
    $c_0 \leftarrow \{0, 1\}^b$
    for $i = 1..l$
        $c_i = f_k(m_i \oplus c_{i-1});$
    output $c = c_0, ..., c_l$
```

Theorem 5.4. *Stateless CBC is IND-CPA secure.*

6 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:

6.1 Real Life Scheme

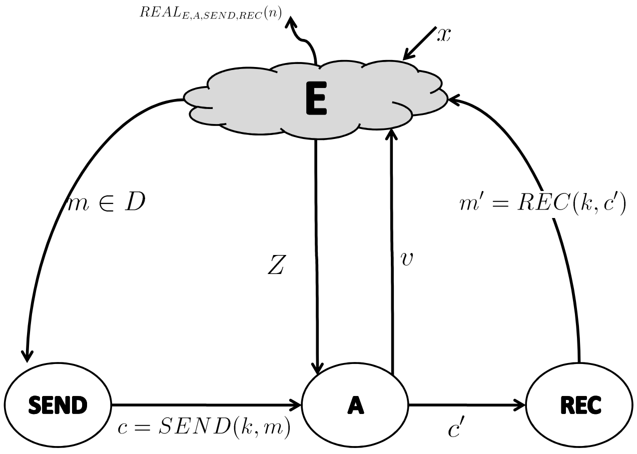

Figure 3: Real secure connection

In the case of plain semantic security we modeled the external world via D, Z, P. Since messages are being chosen interactively, we need a more interactive notion of the external world as a computational entity; we'll call it E, the environment. The real world interaction proceeds as follows:

- A key $k \in \{0,1\}^n$ is chosen and given to $SEND$ and REC
- E gets some external input x
- E sends some value m to $SEND$ and $Z(m)$ to A
- A gets $c = SEND(k, m)$
- A sends v to E and some value c' to the REC
- E gets $m' = REC(k, c')$

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.) Notice that REC is allowed to return an error code instead of m'. See Figure 3. Let $Real_{E,A,SEND,REC}(n)$ denotes the output of E in this game; $Real_{E,A,SEND,REC}$ denotes the ensemble $\{Real_{E,A,SEND,REC}(n)\}_{n \in N}$.

6.2 Ideal Scheme

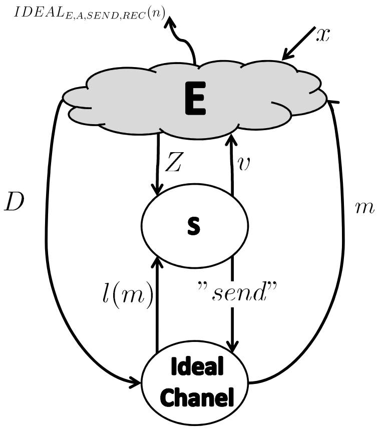

Figure 4: Ideal secure connection

Same as the real-life setting, except that the adversary, S, does not see c and cannot change the messages sent. The ideal scheme proceeds as follows:

- E gets some external input x
- E sends some value m to the ideal channel and $Z(m)$ to S
- S gets $l(m)$
- S sends v to E and potentially also a "send" command that sends m to E.

E repeats this process until at some point it outputs a value. Notice that the adversary 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[1]. See Figure 4. Let $Ideal_{E,A}(n)$ denotes the output of E in this game; $Ideal_{E,A}$ denotes $\{Ideal_{E,A}(n)\}_{n \in N}$.

6.3 "Behaves Like the Ideal Scheme"

The notion is the same as before, except that now D,Z,P are incorporated into one entity, E:

Definition 6.1. *A pair of interactive algorithms ($SEND$, REC) is a secure channels protocol if for any polytime adversary A there exists an adversary S such that for any environment E we have:*

$$Real_{E,A,SEND,REC} \approx_c Ideal_{E,A}$$

[1]Alternatively, one can assume that REC dosen't return a value when it receives a forged message; Accordingly S can decide whether to send the "send" command in the Ideal scheme.

6.4 Realizing Secure Channels

The main idea is to combine encryption and MAC. A "natural" scheme is *Encrypt then MAC (EtM)*: Let $E = (ENC, DEC)$ be an IND-CPA encryption scheme, and let $M = (AUTH, VER)$ be a EU-CMA MAC scheme. Then:

```
$SEND\ (k, m)$:
    Initialize $ctr = 0$
    Treat $k$ as $k = k_e, k_a$
    $c = ENC(k_e, m)$
    $t = AUTH(k_a, c, ctr)$
    output $(c, ctr, t)$
    $ctr++$

$REC\ (k, c, ctr, t)$:
    Treat $k$ as $k = k_e, k_a$
    if ($ctr\_set$ contains $ctr$) abort // If we already received a message with the same counter we abort
    if !($VER(k_a, c, ctr, t)$) abort  // If the verifier rejects the message we abort
    add $ctr$ to $ctr\_set$            // We add the counter we received to the set of previously received counters
    output $DEC(k_e, c)$               // We output the plaintext
```

Theorem 6.2. *If E is an IND-CPA scheme and M is an EU-CMA MAC scheme then $EtM(E, M)$ is a secure channels protocol.*

This is a "natural" scheme since it can be thought of as consisting of two layers: an external MAC shell that gives the equivalent of an “authenticated channel” over which the encryption is transmitted.
What about other combinations of ENC and MAC? e.g. $ENC(k_e, m, MAC(k_a, m, ctr)), ctr$?
$ENC(k_e, m), MAC(k_a, m, ctr), ctr$? Left as food for thought.