Introduction to the theory of secret key cryptography

Andreas Hülsing
Eindhoven University of Technology

16 October 2019

Main primitives of secret key / symmetric cryptography

High-level primitives

Low-level primitives

Main primitives of secret key / symmetric cryptography

High-level primitives

- Secret key encryption (SKE)
  - Provides: Secrecy
  - Applications: File encryption, communication secrecy

Low-level primitives

Main primitives of secret key / symmetric cryptography

High-level primitives

- Secret key encryption (SKE)
  - Provides: Secrecy
  - Applications: File encryption, communication secrecy
- Message authentication codes (MAC)
  - Provides: Integrity & authentication
  - Applications: Secure communication (allows for deniability), secure storage

Low-level primitives

Main primitives of secret key / symmetric cryptography

High-level primitives

- Secret key encryption (SKE)
  - Provides: Secrecy
  - Applications: File encryption, communication secrecy
- Message authentication codes (MAC)
  - Provides: Integrity & authentication
  - Applications: Secure communication (allows for deniability), secure storage

Low-level primitives

- Pseudorandom generator (PRG) / function (PRF)
  - Provides: Pseudorandom behaviour
  - Applications: Replace random bits / functions with deterministic object

Main primitives of secret key / symmetric cryptography

High-level primitives

- Secret key encryption (SKE)
  - Provides: Secrecy
  - Applications: File encryption, communication secrecy
- Message authentication codes (MAC)
  - Provides: Integrity & authentication
  - Applications: Secure communication (allows for deniability), secure storage

Low-level primitives

- Pseudorandom generator (PRG) / function (PRF)
  - Provides: Pseudorandom behaviour
  - Applications: Replace random bits / functions with deterministic object
- Cryptographic hash functions
  - Provides: One-wayness, collision resistance
  - Applications: From digital signatures to password hashing and PoW

Secret key encryption

Secret key encryption (SKE)

Secret key cryptography

Definition (Secret key encryption scheme)

A secret key encryption scheme is a tripple of algorithms $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ and a message or plaintext space $\mathcal{M}$ such that the following holds

- Gen is a probabilistic algorithm that outputs a key k. The output space of Gen is called key space $\mathcal{K}$.
- Enc takes as inputs a key $k \in \mathcal{K}$ and message $m \in \mathcal{M}$, and outputs ciphertext $c = \mathrm{Enc}_k(m)$. The output space of Enc is called ciphertext space $\mathcal{C}$.
- Dec is a deterministic algorithm that takes as inputs a key $k \in \mathcal{K}$ and ciphertext $c \in \mathcal{C}$ and outputs a plaintext $m' \in \mathcal{M} : m' = \mathrm{Dec}_k(c)$.

Correctness: $(\forall k \leftarrow \mathrm{Gen}(), \forall m \in \mathcal{M}) : \mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$

How to define security?

Definition (Perfect secrecy)

A secret key encryption scheme $\mathcal{E} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ with message space $\mathcal{M}$ is perfectly secret if for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $\Pr[C = c] > 0$:

$$\Pr[M = m \mid C = c] = \Pr[M = m]$$

Perfect secrecy considers adversaries $\mathcal{A}$ with $\mathrm{unlimited}$ power.

Is perfect secrecy achievable?

A scheme that is perfectly secret is Vernam's one-time pad (OTP):

Construction (One-time pad)

Let $\mathcal{M} = \{0,1\}^\ell (= \mathcal{K} = \mathcal{C})$, the one-time pad is the encryption scheme consisting of the following three algorithms:

$\mathrm{Gen}()$: *Return* $k \leftarrow_R \{0,1\}^\ell$.

$\mathrm{Enc}_k(m)$: *Return* $c = m \oplus k$.

$\mathrm{Dec}_k(c)$: *Return* $m' = c \oplus k$.

Is perfect secrecy achievable?

A scheme that is perfectly secret is Vernam's one-time pad (OTP):

Construction (One-time pad)

Let $\mathcal{M} = \{0,1\}^\ell (= \mathcal{K} = \mathcal{C})$, the one-time pad is the encryption scheme consisting of the following three algorithms:

$\mathrm{Gen}()$: *Return* $k \leftarrow_R \{0,1\}^\ell$.

$\mathrm{Enc}_k(m)$: *Return* $c = m \oplus k$.

$\mathrm{Dec}_k(c)$: *Return* $m' = c \oplus k$.

Correctness

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = (m \oplus k) \oplus k = m$$

Is perfect secrecy achievable?

A scheme that is perfectly secret is Vernam's one-time pad (OTP):

Construction (One-time pad)

Let $\mathcal{M} = \{0,1\}^\ell (= \mathcal{K} = \mathcal{C})$, the one-time pad is the encryption scheme consisting of the following three algorithms:

$\mathrm{Gen}()$: *Return* $k \leftarrow_R \{0,1\}^\ell$.

$\mathrm{Enc}_k(m)$: *Return* $c = m \oplus k$.

$\mathrm{Dec}_k(c)$: *Return* $m' = c \oplus k$.

Correctness

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = (m \oplus k) \oplus k = m$$

Main observation behind security proof

For every pair (m, c) of message and ciphertext there exists exactly one key that encrypts m as c.

Is perfect secrecy **efficiently** achievable?

Theorem

Let $\mathcal{E}$ be a perfectly secret encryption scheme over message space $\mathcal{M}$, and let $\mathcal{K}$ be the key space determined by Gen. *Then*

$$|\mathcal{K}| \geq |\mathcal{M}|.$$

Is perfect secrecy **efficiently** achievable?

Theorem

Let $\mathcal{E}$ be a perfectly secret encryption scheme over message space $\mathcal{M}$, and let $\mathcal{K}$ be the key space determined by Gen. *Then*

$$|\mathcal{K}| \geq |\mathcal{M}|.$$

Proof sketch

Assume $|\mathcal{K}| < |\mathcal{M}|$.

1. An arbitrary ciphertext c can only decrypt to $\leq |\mathcal{K}|$ different messages.
2. Consequently, there exist messages m such that $\Pr[M = m \mid C = c] = 0$.
3. If we choose the uniform distribution as message distribution $\Pr[M = m] > 0$.

Hence, $\mathcal{E}$ is not perfectly secure.

Back to square 1: How to define security?

Consider security against efficient (= computationally bounded [= polytime]) adversaries.

Experiment- / game-based security definitions: We define a game that is played by the adversary and analyze its success probability.

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.
Simulation-based security: $\mathcal{A}$ is compared to simulator $\mathcal{S}$ which plays in a slightly different 'experiment' (real VS ideal).

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.
Simulation-based security: $\mathcal{A}$ is compared to simulator $\mathcal{S}$ which plays in a slightly different 'experiment' (real VS ideal).

$\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$:

$\mathcal{A}$ chooses a challenge template:

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.
Simulation-based security: $\mathcal{A}$ is compared to simulator $\mathcal{S}$ which plays in a slightly different 'experiment' (real VS ideal).

$\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$:

$\mathcal{A}$ chooses a challenge template:

- a message distribution X on plaintext space $\mathcal{M}$,

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.
Simulation-based security: $\mathcal{A}$ is compared to simulator $\mathcal{S}$ which plays in a slightly different 'experiment' (real VS ideal).

$\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$:

$\mathcal{A}$ chooses a challenge template:

- a message distribution X on plaintext space $\mathcal{M}$,
- an advice function $h : \mathcal{M} \to \mathbb{N}$,

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.
Simulation-based security: $\mathcal{A}$ is compared to simulator $\mathcal{S}$ which plays in a slightly different 'experiment' (real VS ideal).

$\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$:

$\mathcal{A}$ chooses a challenge template:

- a message distribution X on plaintext space $\mathcal{M}$,
- an advice function $h : \mathcal{M} \to \mathbb{N}$,
- a target function $f : \mathcal{M} \to \mathbb{N}$.

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.
Simulation-based security: $\mathcal{A}$ is compared to simulator $\mathcal{S}$ which plays in a slightly different 'experiment' (real VS ideal).

$\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$:

$\mathcal{A}$ chooses a challenge template:

- a message distribution X on plaintext space $\mathcal{M}$,
- an advice function $h : \mathcal{M} \to \mathbb{N}$,
- a target function $f : \mathcal{M} \to \mathbb{N}$.

x is sampled from X and $\mathcal{A}$ receives $(\mathrm{Enc}_k(x), h(x))$. $\mathcal{A}$ succeeds if $\mathcal{A}(\mathrm{Enc}_k(x), h(x)) = f(x)$.

Attempt 1: Semantic security

Intuition: Everything adversary $\mathcal{A}$ learns about m knowing c, one could have learned without knowing c.
Simulation-based security: $\mathcal{A}$ is compared to simulator $\mathcal{S}$ which plays in a slightly different 'experiment' (real VS ideal).

$\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$:

$\mathcal{A}$ chooses a challenge template:

- a message distribution X on plaintext space $\mathcal{M}$,
- an advice function $h : \mathcal{M} \to \mathbb{N}$,
- a target function $f : \mathcal{M} \to \mathbb{N}$.

x is sampled from X and $\mathcal{A}$ receives $(\mathrm{Enc}_k(x), h(x))$. $\mathcal{A}$ succeeds if $\mathcal{A}(\mathrm{Enc}_k(x), h(x)) = f(x)$.
($\mathcal{S}$ only receives $h(x)$. $\mathcal{S}$ succeeds if $\mathcal{S}(h(x)) = f(x)$.)

Attempt 1: Semantic security

Definition (Semantic Security (SEM))

A secret key encryption scheme has semantic security if for any efficient adversary $\mathcal{A}$ there exists an efficient simulator $\mathcal{S}$ such that their probabilites of success playing $\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$ are negligibly close to each other.

Attempt 1: Semantic security

Definition (Semantic Security (SEM))

A secret key encryption scheme has semantic security if for any efficient adversary $\mathcal{A}$ there exists an efficient simulator $\mathcal{S}$ such that their probabilites of success playing $\mathrm{Exp}^{\mathrm{SEM}}_{\mathcal{E},\mathcal{A}}(n)$ are negligibly close to each other.

For unbounded adversaries this is equivalent to perfect secrecy.

This definition is cumbersome to work with!

Attempt 2: Indistinguishable ciphertext security

$\mathsf{Exp}^{\mathsf{IND}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$

Attempt 2: Indistinguishable ciphertext security

$\mathrm{Exp}^{\mathrm{IND}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$

Attempt 2: Indistinguishable ciphertext security

$\mathrm{Exp}^{\mathrm{IND}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$
3. $b \leftarrow_R \{0,1\}, c \leftarrow \mathrm{Enc}_k(m_b)$

Attempt 2: Indistinguishable ciphertext security

$\mathrm{Exp}^{\mathrm{IND}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$
3. $b \leftarrow_R \{0,1\}, c \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}(c)$

Attempt 2: Indistinguishable ciphertext security

$\mathrm{Exp}^{\mathrm{IND}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$
3. $b \leftarrow_R \{0,1\}, c \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}(c)$
5. Output 1 if $b' = b$, otherwise 0

Attempt 2: Indistinguishable ciphertext security

$\mathrm{Exp}^{\mathrm{IND}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$
3. $b \leftarrow_R \{0,1\}, c \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}(c)$
5. Output 1 if $b' = b$, otherwise 0

Definition (Indistinguishable ciphertexts (IND))

A secret key encryption scheme $\mathcal{E}$ has indistinguishable ciphertexts if for all efficient adversaries $\mathcal{A}$ their advantage ε in winning above game is negligible

$$\Pr\left[\mathrm{Exp}^{\mathrm{IND}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] = \frac{1}{2} + \varepsilon.$$

This definition is a lot easier to work with and equivalent to SEM!

Is IND efficiently achievable?

We first need tooling.

Definition (Pseudorandom generator (PRG))

Let ℓ be a polynomial and let G be a deterministic, efficient algorithm that implements a function $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$. We say G is a secure PRG if the following two conditions hold:

Is IND efficiently achievable?

We first need tooling.

Definition (Pseudorandom generator (PRG))

Let ℓ be a polynomial and let G be a deterministic, efficient algorithm that implements a function $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$. We say G is a secure PRG if the following two conditions hold:

1. **Expansion:** For every n it holds that $\ell(n) > n$.

Is IND efficiently achievable?

We first need tooling.

Definition (Pseudorandom generator (PRG))

Let ℓ be a polynomial and let G be a deterministic, efficient algorithm that implements a function $\mathsf{G} : \{0,1\}^n \to \{0,1\}^{\ell(n)}$. We say G is a secure PRG if the following two conditions hold:

1. **Expansion:** For every n it holds that $\ell(n) > n$.
2. **Pseudorandomness:** For all efficient distinguishers $\mathcal{D}$ the advantage ε distinguishing outputs of G from random is negligible, where

$$\varepsilon = \left| \Pr_{r \leftarrow_R \{0,1\}^{\ell(n)}} [\mathcal{D}(r) = 1] - \Pr_{s \leftarrow_R \{0,1\}^n} [\mathcal{D}(\mathsf{G}(s)) = 1] \right| .$$

Is IND efficiently achievable?

We first need tooling.

Definition (Pseudorandom generator (PRG))

Let ℓ be a polynomial and let G be a deterministic, efficient algorithm that implements a function $\mathsf{G} : \{0,1\}^n \to \{0,1\}^{\ell(n)}$. We say G is a secure PRG if the following two conditions hold:

1. **Expansion:** For every n it holds that $\ell(n) > n$.
2. **Pseudorandomness:** For all efficient distinguishers $\mathcal{D}$ the advantage ε distinguishing outputs of G from random is negligible, where

$$\varepsilon = \left| \Pr_{r \leftarrow_R \{0,1\}^{\ell(n)}} [\mathcal{D}(r) = 1] - \Pr_{s \leftarrow_R \{0,1\}^n} [\mathcal{D}(\mathsf{G}(s)) = 1] \right| .$$

PRG's exist if one-way functions exist. Will see examples later.

Is IND efficiently achievable?

Construction (PRG-ENC)

Let $n \in \mathbb{N}$ be the security parameter, let $\mathcal{M} = \{0,1\}^{\ell(n)} (= \mathcal{C})$, and let G be a PRG as defined above. The PRG-ENC encryption scheme consists of the following three algorithms:

$\mathrm{Gen}(1^n)$: *Return* $k \leftarrow_R \{0,1\}^n$.

$\mathrm{Enc}_k(m)$: *Return* $c = m \oplus \mathsf{G}(k)$.

$\mathrm{Dec}_k(c)$: *Return* $m' = c \oplus \mathsf{G}(k)$.

Is IND efficiently achievable?

Construction (PRG-ENC)

Let $n \in \mathbb{N}$ be the security parameter, let $\mathcal{M} = \{0,1\}^{\ell(n)} (= \mathcal{C})$, and let G be a PRG as defined above. The PRG-ENC encryption scheme consists of the following three algorithms:

$\mathrm{Gen}(1^n)$: *Return* $k \leftarrow_R \{0,1\}^n$.

$\mathrm{Enc}_k(m)$: *Return* $c = m \oplus \mathsf{G}(k)$.

$\mathrm{Dec}_k(c)$: *Return* $m' = c \oplus \mathsf{G}(k)$.

Correctness

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = (m \oplus \mathsf{G}(k)) \oplus \mathsf{G}(k) = m$$

PRG-ENC is IND secure

Proof by reduction. If there exists $\mathcal{A}$ that can distinguish ciphertexts of PRG-ENC in time t with advantage ε then the following algorithm $\mathcal{D}$ runs in time $\approx t$ and succeeds in distinguishing G with advantage $\varepsilon' = \varepsilon$.

PRG-ENC is IND secure

Proof by reduction. If there exists $\mathcal{A}$ that can distinguish ciphertexts of PRG-ENC in time t with advantage ε then the following algorithm $\mathcal{D}$ runs in time $\approx t$ and succeeds in distinguishing G with advantage $\varepsilon' = \varepsilon$.

Construction (Distinguisher $\mathcal{D}$)

Given as input a string $w \in \{0,1\}^{\ell(n)}$*:*

1. *Run* $m_0, m_1 \leftarrow \mathcal{A}(1^n)$
2. *Set* $b \leftarrow_R \{0,1\}, c = m_b \oplus w$
3. *Run* $b' \leftarrow \mathcal{A}(c)$
4. *Return 1 if* $b = b'$*, otherwise 0.*

Advantage of $\mathcal{D}$

Construction (Distinguisher $\mathcal{D}$)

Given as input a string $w \in \{0,1\}^{\ell(n)}$*:*

1. *Run* $m_0, m_1 \leftarrow \mathcal{A}(1^n)$
2. *Set* $b \leftarrow_R \{0,1\}, c = m_b \oplus w$
3. *Run* $b' \leftarrow \mathcal{A}(c)$
4. *Return 1 if* $b = b'$*, otherwise 0.*

$$\varepsilon' = |\Pr[\mathcal{D}(r) = 1] - \Pr[\mathcal{D}(G(s)) = 1]|$$

Advantage of $\mathcal{D}$

Construction (Distinguisher $\mathcal{D}$)

Given as input a string $w \in \{0,1\}^{\ell(n)}$*:*

1. *Run* $m_0, m_1 \leftarrow \mathcal{A}(1^n)$
2. *Set* $b \leftarrow_R \{0,1\}, c = m_b \oplus w$
3. *Run* $b' \leftarrow \mathcal{A}(c)$
4. *Return 1 if* $b = b'$*, otherwise 0.*

$$\varepsilon' = |\Pr[\mathcal{D}(r) = 1] - \Pr[\mathcal{D}(G(s)) = 1]|$$

$$\Pr[\mathcal{D}(r) = 1] = \Pr\left[\mathrm{Exp}^{\mathrm{IND}}_{\mathrm{OTP},\mathcal{A}}(n) = 1\right] = \frac{1}{2}$$

Advantage of $\mathcal{D}$

Construction (Distinguisher $\mathcal{D}$)

Given as input a string $w \in \{0,1\}^{\ell(n)}$*:*

1. *Run* $m_0, m_1 \leftarrow \mathcal{A}(1^n)$
2. *Set* $b \leftarrow_R \{0,1\}, c = m_b \oplus w$
3. *Run* $b' \leftarrow \mathcal{A}(c)$
4. *Return 1 if* $b = b'$*, otherwise 0.*

$$\varepsilon' = |\Pr[\mathcal{D}(r) = 1] - \Pr[\mathcal{D}(G(s)) = 1]|$$

$$\Pr[\mathcal{D}(r) = 1] = \Pr\left[\mathrm{Exp}^{\mathrm{IND}}_{\mathrm{OTP},\mathcal{A}}(n) = 1\right] = \frac{1}{2}$$

$$\Pr[\mathcal{D}(G(s)) = 1] = \Pr\left[\mathrm{Exp}^{\mathrm{IND}}_{\mathrm{PRG-ENC},\mathcal{A}}(n) = 1\right] = \frac{1}{2} + \varepsilon$$

Advantage of $\mathcal{D}$

Construction (Distinguisher $\mathcal{D}$)

Given as input a string $w \in \{0,1\}^{\ell(n)}$*:*

1. *Run* $m_0, m_1 \leftarrow \mathcal{A}(1^n)$
2. *Set* $b \leftarrow_R \{0,1\}, c = m_b \oplus w$
3. *Run* $b' \leftarrow \mathcal{A}(c)$
4. *Return 1 if* $b = b'$*, otherwise 0.*

$$\varepsilon' = |\Pr[\mathcal{D}(r) = 1] - \Pr[\mathcal{D}(G(s)) = 1]|$$

$$\Pr[\mathcal{D}(r) = 1] = \Pr\left[\mathrm{Exp}^{\mathrm{IND}}_{\mathrm{OTP},\mathcal{A}}(n) = 1\right] = \frac{1}{2}$$

$$\Pr[\mathcal{D}(G(s)) = 1] = \Pr\left[\mathrm{Exp}^{\mathrm{IND}}_{\mathrm{PRG-ENC},\mathcal{A}}(n) = 1\right] = \frac{1}{2} + \varepsilon$$

$$\varepsilon' = \left|\frac{1}{2} - \left(\frac{1}{2} + \varepsilon\right)\right| = \varepsilon$$

PRG-ENC is IND secure

Theorem

If there exists $\mathcal{A}$ that can distinguish ciphertexts of PRG-ENC in time t with advantage ε then the algorithm $\mathcal{D}$ from above runs in time $\approx t$ and succeeds in breaking G with advantage $\varepsilon' = \varepsilon$.
Hence, if G is a secure PRG, then PRG-ENC has indistinguishable ciphertexts.

What did we achieve?

- SEM, IND, and perfect secrecy define $\mathcal{A}$'s goal

What did we achieve?

- SEM, IND, and perfect secrecy define $\mathcal{A}$'s goal
- What about $\mathcal{A}$'s attack capabilities?

What did we achieve?

- SEM, IND, and perfect secrecy define $\mathcal{A}$'s goal
- What about $\mathcal{A}$'s attack capabilities?
- In this sense they are unrealistic single message notions.

Is this realistic?

Or rather this.

What can $\mathcal{A}$ learn?

- Often messages follow known format (MIME, HTML, XML,...).

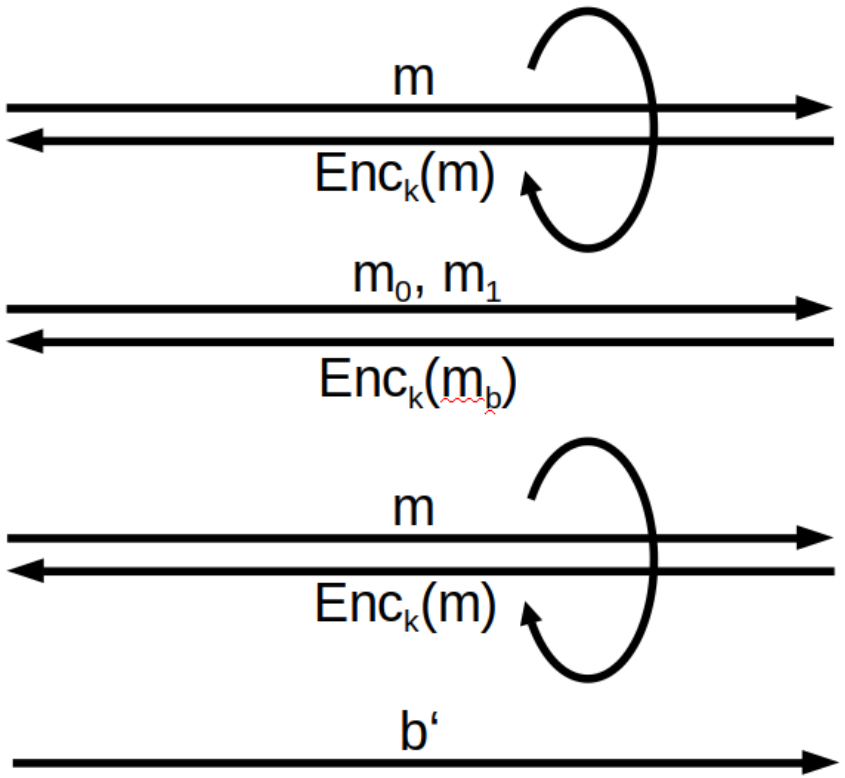
What can $\mathcal{A}$ learn?

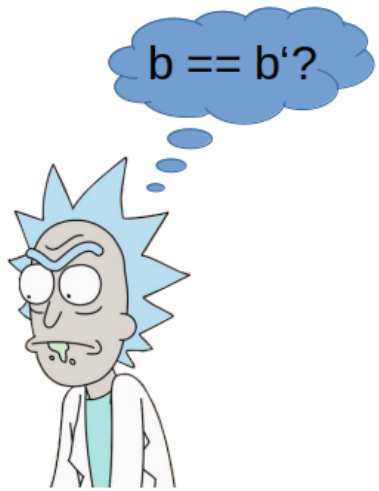
- Often messages follow known format (MIME, HTML, XML,...).
- Often parts of messages are guessable:
  - "To whom it may concern,"
  - "Dear [Recipient],"
  - "Best regards, \n [Sender]"
  - "Cheers, \n [Sender]"

What can $\mathcal{A}$ learn?

- Often messages follow known format (MIME, HTML, XML,...).
- Often parts of messages are guessable:
  - "To whom it may concern,"
  - "Dear [Recipient],"
  - "Best regards, \n [Sender]"
  - "Cheers, \n [Sender]"
- Want to model the worst case: Let $\mathcal{A}$ choose messages that get encrypted!

IND under chosen plaintext attacks (IND-CPA)





IND under chosen plaintext attacks (IND-CPA).

$\mathrm{Exp}^{\mathrm{IND-CPA}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot)}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$
3. $b \leftarrow_R \{0,1\}, c \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot)}(c)$
5. Output 1 if $b' = b$, otherwise 0

IND under chosen plaintext attacks (IND-CPA).

$\mathrm{Exp}^{\mathrm{IND-CPA}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot)}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$
3. $b \leftarrow_R \{0,1\}, c \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot)}(c)$
5. Output 1 if $b' = b$, otherwise 0

Definition (IND-CPA)

A secret key encryption scheme $\mathcal{E}$ has indistinguishable ciphertexts under chosen plaintext attacks if for all efficient adversaries $\mathcal{A}$ their advantage ε in winning above game is negligible

$$\Pr\left[\mathrm{Exp}^{\mathrm{IND-CPA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] \leq \frac{1}{2} + \varepsilon.$$

IND under chosen plaintext attacks (IND-CPA).

$\mathrm{Exp}^{\mathrm{IND-CPA}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot)}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0| = |m_1|$
3. $b \leftarrow_R \{0,1\}, c \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot)}(c)$
5. Output 1 if $b' = b$, otherwise 0

Definition (IND-CPA)

A secret key encryption scheme $\mathcal{E}$ has indistinguishable ciphertexts under chosen plaintext attacks if for all efficient adversaries $\mathcal{A}$ their advantage ε in winning above game is negligible

$$\Pr\left[\mathrm{Exp}^{\mathrm{IND-CPA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] \leq \frac{1}{2} + \varepsilon.$$

Note: This definition is equivalent to SEM-CPA.

IND-CPA secure SKE

- Is the one-time pad IND-CPA-secure?

IND-CPA secure SKE

- Is the one-time pad IND-CPA-secure?
- What about PRG-ENC?

IND-CPA secure SKE

- Is the one-time pad IND-CPA-secure?
- What about PRG-ENC?

Theorem

A deterministic encryption scheme cannot be IND-CPA secure.

IND-CPA secure SKE

- Is the one-time pad IND-CPA-secure?
- What about PRG-ENC?

Theorem

A deterministic encryption scheme cannot be IND-CPA secure.

Proof idea.

Send m_0 to $\mathrm{Enc}_k(\cdot)$ and compare result with challenge ciphertext. □

Pseudorandom function families

A keyed function is a two input function $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ where the first input is called the key and denoted k. We will write $\mathsf{F}_k(x) \stackrel{def}{=} \mathsf{F}(k, x)$.

Pseudorandom function families

A keyed function is a two input function $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ where the first input is called the key and denoted k. We will write $\mathsf{F}_k(x) \stackrel{def}{=} \mathsf{F}(k, x)$.

Definition (Pseudorandom function family (PRF))

Let $\mathsf{F} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an efficient, length-preserving, keyed function. We say F is a pseudorandom function if for all efficient distinguishers $\mathcal{D}$ the distinguishing advantage ε is negligible, where

$$\varepsilon = \left| \Pr_{k \leftarrow_R \{0,1\}^n} \left[\mathcal{D}^{\mathsf{F}_k(\cdot)}(1^n) = 1 \right] - \Pr_{f_n \leftarrow_R \mathsf{FUNC}_n} \left[\mathcal{D}^{f_n(\cdot)}(1^n) = 1 \right] \right| .$$

Pseudorandom function families

A keyed function is a two input function $\mathsf{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ where the first input is called the key and denoted k. We will write $\mathsf{F}_k(x) \stackrel{def}{=} \mathsf{F}(k, x)$.

Definition (Pseudorandom function family (PRF))

Let $\mathsf{F} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an efficient, length-preserving, keyed function. We say F is a pseudorandom function if for all efficient distinguishers $\mathcal{D}$ the distinguishing advantage ε is negligible, where

$$\varepsilon = \left| \Pr_{k \leftarrow_R \{0,1\}^n} \left[\mathcal{D}^{\mathsf{F}_k(\cdot)}(1^n) = 1 \right] - \Pr_{f_n \leftarrow_R \mathsf{FUNC}_n} \left[\mathcal{D}^{f_n(\cdot)}(1^n) = 1 \right] \right| .$$

PRF's exist if PRG's exist [GGM'84]. For length doubling PRG G define

$$\mathsf{F}_k(x) \stackrel{def}{=} \mathsf{G}\left(\ldots \mathsf{G}\left(\mathsf{G}(k)_{x_1} \right)_{x_2} \ldots \right)_{x_n} .$$

Pseudorandom permutation families

Formal model for block ciphers is PRP.

Definition (Pseudorandom permutation family (PRP))

Let $n \in \mathbb{N}$ be the security parameter,
$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an efficient, length-preserving, keyed permutation. We say F is a family of pseudorandom permutations (PRP) if for all efficient distinguishers $\mathcal{D}$ the distinguishing advantage ε is negligible, where

$$\varepsilon = \left| \Pr_{k \leftarrow_R \{0,1\}^n} \left[\mathcal{D}^{F_k(\cdot), F_k^{-1}(\cdot)}(1^n) = 1 \right] - \Pr_{f_n \leftarrow_R \mathrm{PERM}_n} \left[\mathcal{D}^{f_n(\cdot), f_n^{-1}(\cdot)}(1^n) = 1 \right] \right|,$$

where PERM_n denotes the set of all permutations over $\{0,1\}^n$.

A PRP is a PRF (Switching-Lemma) but not vice-versa.

IND-CPA-secure SKE

Construction (PRF-ENC)

Let $n \in \mathbb{N}$ be the security parameter, let $\mathcal{M} = \{0,1\}^n (= \mathcal{C} = \mathcal{K})$, and let F be a length-preserving PRF as defined above. The PRF-ENC encryption scheme consists of the following three algorithms:

$\mathrm{Gen}(1^n)$: *Return $k \leftarrow_R \{0,1\}^n$.*

$\mathrm{Enc}_k(m)$: *Sample $r \leftarrow_R \{0,1\}^n$, compute $\bar{c} = m \oplus \mathsf{F}_k(r)$, and return $c = \langle r, \bar{c} \rangle$.*

$\mathrm{Dec}_k(c)$: *Parse c as $\langle r, \bar{c} \rangle$. Return $m' = \bar{c} \oplus \mathsf{F}_k(r)$.*

IND-CPA-secure SKE

Construction (PRF-ENC)

Let $n \in \mathbb{N}$ be the security parameter, let $\mathcal{M} = \{0,1\}^n (= \mathcal{C} = \mathcal{K})$, and let F be a length-preserving PRF as defined above. The PRF-ENC encryption scheme consists of the following three algorithms:

$\mathrm{Gen}(1^n)$: *Return $k \leftarrow_R \{0,1\}^n$.*

$\mathrm{Enc}_k(m)$: *Sample $r \leftarrow_R \{0,1\}^n$, compute $\bar{c} = m \oplus \mathsf{F}_k(r)$, and return $c = \langle r, \bar{c} \rangle$.*

$\mathrm{Dec}_k(c)$: *Parse c as $\langle r, \bar{c} \rangle$. Return $m' = \bar{c} \oplus \mathsf{F}_k(r)$.*

Correctness

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = (m \oplus \mathsf{F}_k(r)) \oplus \mathsf{F}_k(r) = m$$

PRF-ENC is IND-CPA secure

Proof idea. Similar to PRG-ENC. Given $\mathcal{A}$ that breaks IND-CPA of PRF-ENC in time t, with advantage ε then the following algorithm $\mathcal{D}$ runs in time $\approx t$ and succeeds in distinguishing F with advantage $\varepsilon' \approx \varepsilon$.

PRF-ENC is IND-CPA secure

Proof idea. Similar to PRG-ENC. Given $\mathcal{A}$ that breaks IND-CPA of PRF-ENC in time t, with advantage ε then the following algorithm $\mathcal{D}$ runs in time $\approx t$ and succeeds in distinguishing F with advantage $\varepsilon' \approx \varepsilon$.

Construction (Distinguisher $\mathcal{D}$)

Given access to oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$*:*

1. *Run* $m_0, m_1 \leftarrow \mathcal{A}^{\mathrm{Enc}'(\cdot)}(1^n)$
2. *Set* $b \leftarrow_R \{0,1\}, r^* \leftarrow_R \{0,1\}^n, \bar{c}^* = m_b \oplus \mathcal{O}(r^*)$
3. *Run* $b' \leftarrow \mathcal{A}^{\mathrm{Enc}'(\cdot)}(\langle r^*, \bar{c}^* \rangle)$
4. *Return 1 if* $b = b'$*, otherwise 0*

where $\mathrm{Enc}'(\cdot)$ *computes* $r \leftarrow_R \{0,1\}^n, \bar{c} = m_b \oplus \mathcal{O}(r)$ *and returns* $\langle r, \bar{c} \rangle$.

Advantage of $\mathcal{D}$

Construction (Distinguisher $\mathcal{D}$)

Given access to oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$*:*

2. *Set* $b \leftarrow_R \{0,1\}, r^* \leftarrow_R \{0,1\}^n, \bar{c}^* = m_b \oplus \mathcal{O}(r^*)$

where $\mathrm{Enc}'(\cdot)$ *computes* $r \leftarrow_R \{0,1\}^n, \bar{c} = m_b \oplus \mathcal{O}(r)$ *and returns* $\langle r, \bar{c} \rangle$.

$$\varepsilon' = \left| \Pr_{k \leftarrow_R \{0,1\}^n} \left[\mathcal{D}^{\mathsf{F}_k(\cdot)}(1^n) = 1 \right] - \Pr_{f_n \leftarrow_R \mathsf{FUNC}_n} \left[\mathcal{D}^{f_n(\cdot)}(1^n) = 1 \right] \right|$$

Advantage of $\mathcal{D}$

Construction (Distinguisher $\mathcal{D}$)

Given access to oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$*:*

2. *Set* $b \leftarrow_R \{0,1\}, r^* \leftarrow_R \{0,1\}^n, \bar{c}^* = m_b \oplus \mathcal{O}(r^*)$

where $\mathrm{Enc}'(\cdot)$ *computes* $r \leftarrow_R \{0,1\}^n, \bar{c} = m_b \oplus \mathcal{O}(r)$ *and returns* $\langle r, \bar{c} \rangle$.

$$\varepsilon' = \left| \Pr_{k \leftarrow_R \{0,1\}^n} \left[\mathcal{D}^{\mathsf{F}_k(\cdot)}(1^n) = 1 \right] - \Pr_{f_n \leftarrow_R \mathsf{FUNC}_n} \left[\mathcal{D}^{f_n(\cdot)}(1^n) = 1 \right] \right|$$

$$= \left| \Pr \left[\mathsf{Exp}^{\mathsf{IND-CPA}}_{\mathrm{PRF-ENC},\mathcal{A}}(n) = 1 \right] - \Pr \left[\mathsf{Exp}^{\mathsf{IND-CPA}}_{\widetilde{\mathrm{PRF-ENC}},\mathcal{A}}(n) = 1 \right] \right|$$

Advantage of $\mathcal{D}$

Construction (Distinguisher $\mathcal{D}$)

Given access to oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$*:*

2. *Set* $b \leftarrow_R \{0,1\}, r^* \leftarrow_R \{0,1\}^n, \bar{c}^* = m_b \oplus \mathcal{O}(r^*)$

where $\mathrm{Enc}'(\cdot)$ *computes* $r \leftarrow_R \{0,1\}^n, \bar{c} = m_b \oplus \mathcal{O}(r)$ *and returns* $\langle r, \bar{c} \rangle$.

$$
\begin{aligned}
\varepsilon' &= \left| \Pr_{k \leftarrow_R \{0,1\}^n} \left[\mathcal{D}^{\mathsf{F}_k(\cdot)}(1^n) = 1 \right] - \Pr_{f_n \leftarrow_R \mathsf{FUNC}_n} \left[\mathcal{D}^{f_n(\cdot)}(1^n) = 1 \right] \right| \\
&= \left| \Pr \left[\mathrm{Exp}^{\mathsf{IND-CPA}}_{\mathrm{PRF-ENC},\mathcal{A}}(n) = 1 \right] - \Pr \left[\mathrm{Exp}^{\mathsf{IND-CPA}}_{\widetilde{\mathrm{PRF-ENC}},\mathcal{A}}(n) = 1 \right] \right| \\
&= \left| \frac{1}{2} + \varepsilon - \left(\frac{1}{2} + \frac{q}{2^n} \right) \right| = \left| \varepsilon - \frac{q}{2^n} \right|
\end{aligned}
$$

PRF-ENC is IND-CPA secure

Theorem

If there exists $\mathcal{A}$ that can distinguish ciphertexts of PRF-ENC during a CPA-experiment in time t with advantage ε then the algorithm $\mathcal{D}$ from above runs in time $\approx t$ and succeeds in breaking F *with advantage $\varepsilon' \geq \varepsilon - q/2^n$.*
Hence, if F *is a secure PRF, then PRF-ENC has indistinguishable ciphertexts under chosen plaintext attacks.*

Arbitrary length messages

PRF-ENC only works for n-bit messages.

Arbitrary length messages

PRF-ENC only works for n-bit messages.

Can repeat fixed-length scheme:
For ℓn-bit message $m = (m_1 \| m_2 \| \ldots \| m_\ell)$ ciphertext is

$$c = \langle r_1, \mathsf{F}_k(r_1) \oplus m_1, r_2, F_k(r_2) \oplus m_2, \ldots, r_\ell, F_k(r_\ell) \oplus m_\ell \rangle$$

Arbitrary length messages

PRF-ENC only works for n-bit messages.

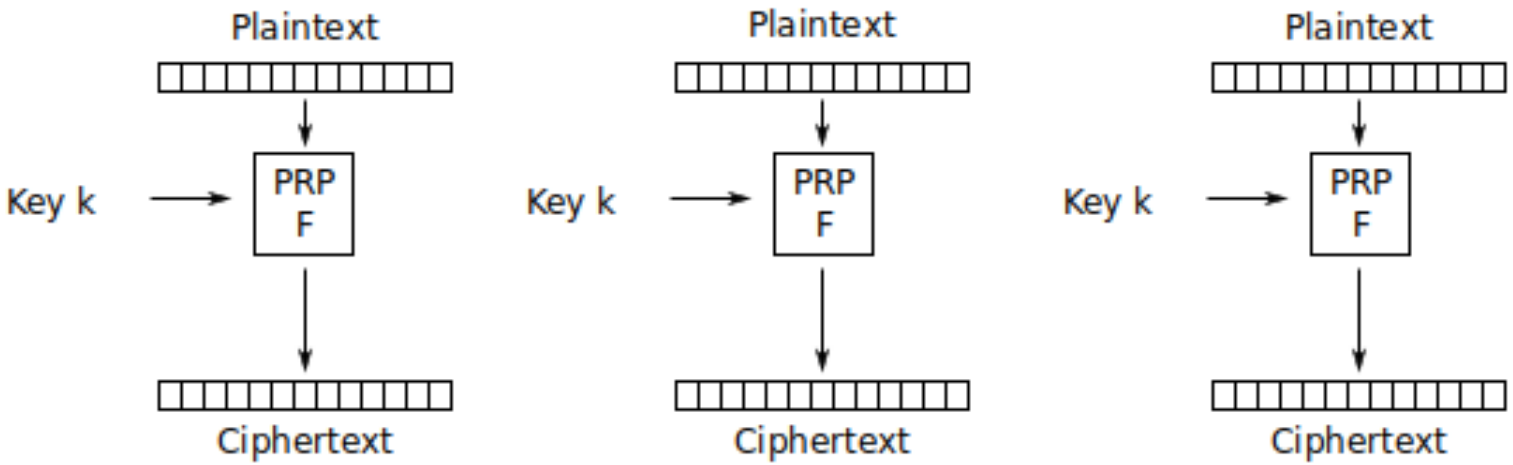
Can repeat fixed-length scheme:
For ℓn-bit message $m = (m_1 \| m_2 \| \ldots \| m_\ell)$ ciphertext is

$$c = \langle r_1, \mathsf{F}_k(r_1) \oplus m_1, r_2, F_k(r_2) \oplus m_2, \ldots, r_\ell, F_k(r_\ell) \oplus m_\ell \rangle$$

Pretty inefficient!

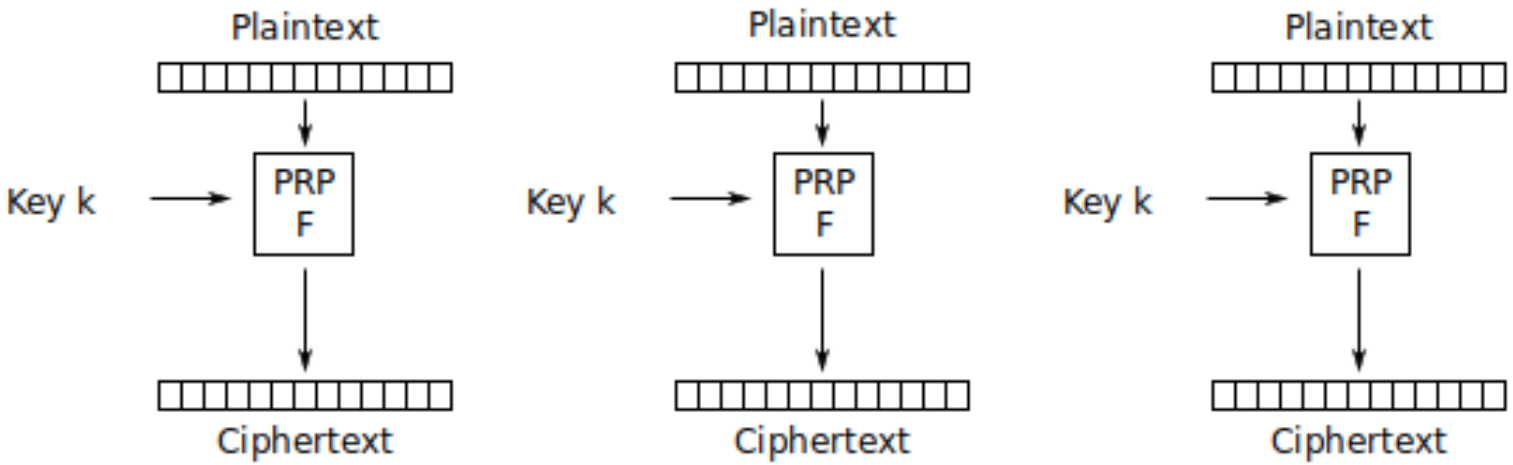
Solution: Modes of operation

Electronic code book mode (ECB)



Electronic Codebook (ECB) mode encryption

Electronic code book mode (ECB)

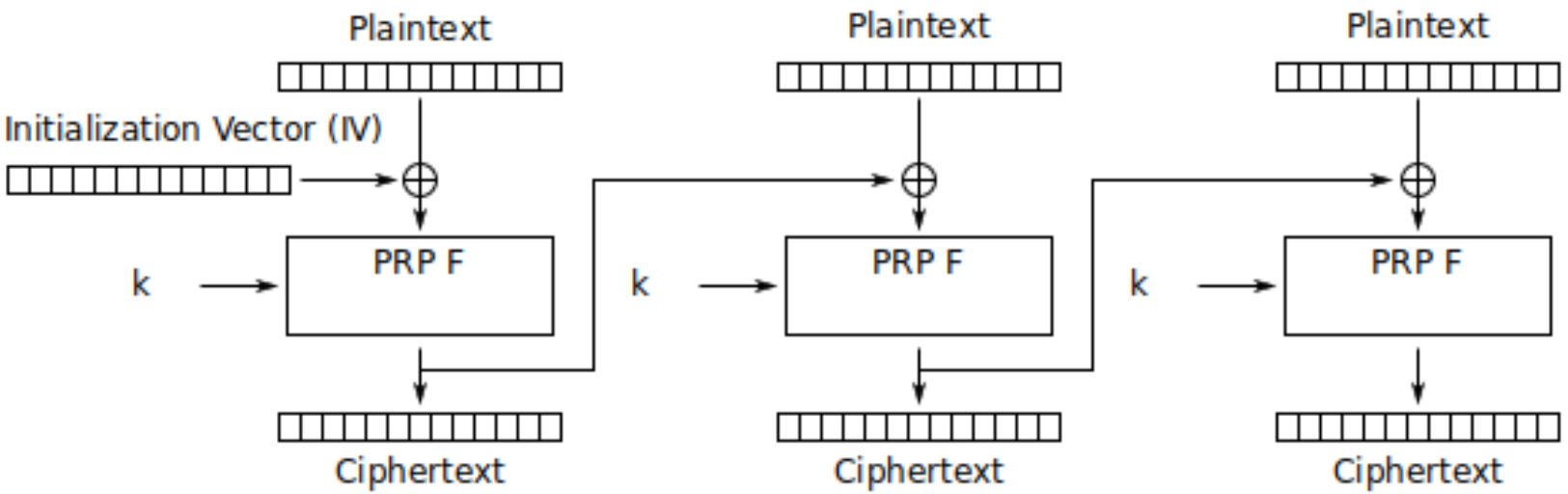


Electronic Codebook (ECB) mode encryption

Deterministic! Even worse, not even IND for single message attacks!

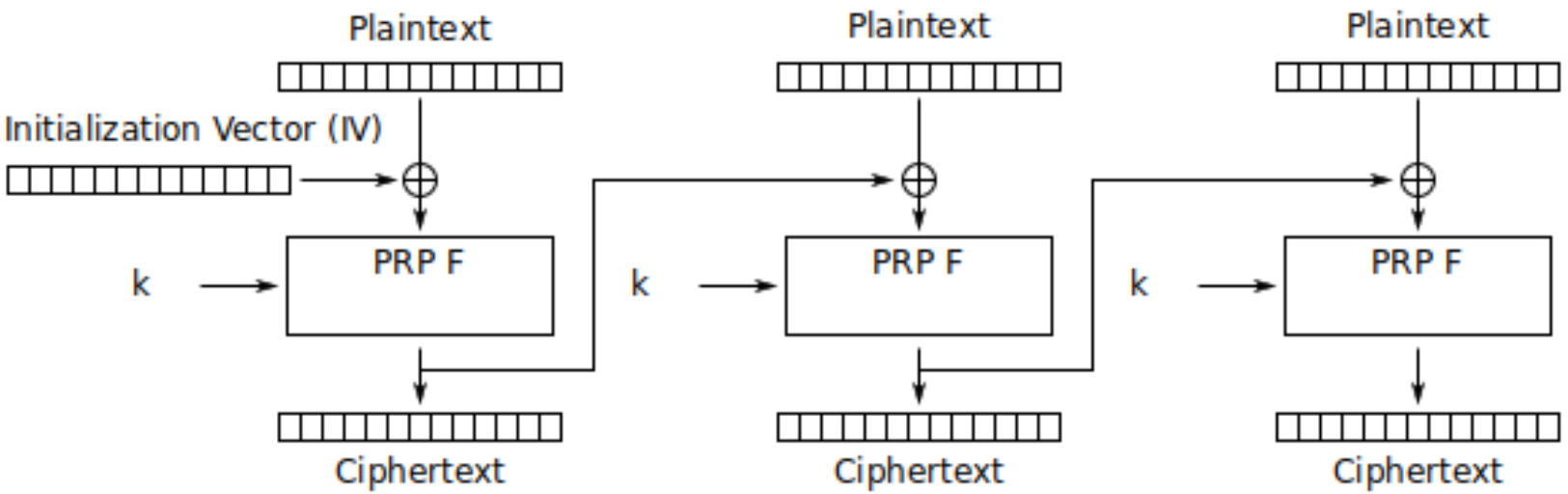
(Consider $m_0 = m\|m; m_1 = m\|m'$ for $m, m' \in \{0,1\}^n$)

Cipher block chaining mode (CBC)



Cipher Block Chaining (CBC) mode encryption

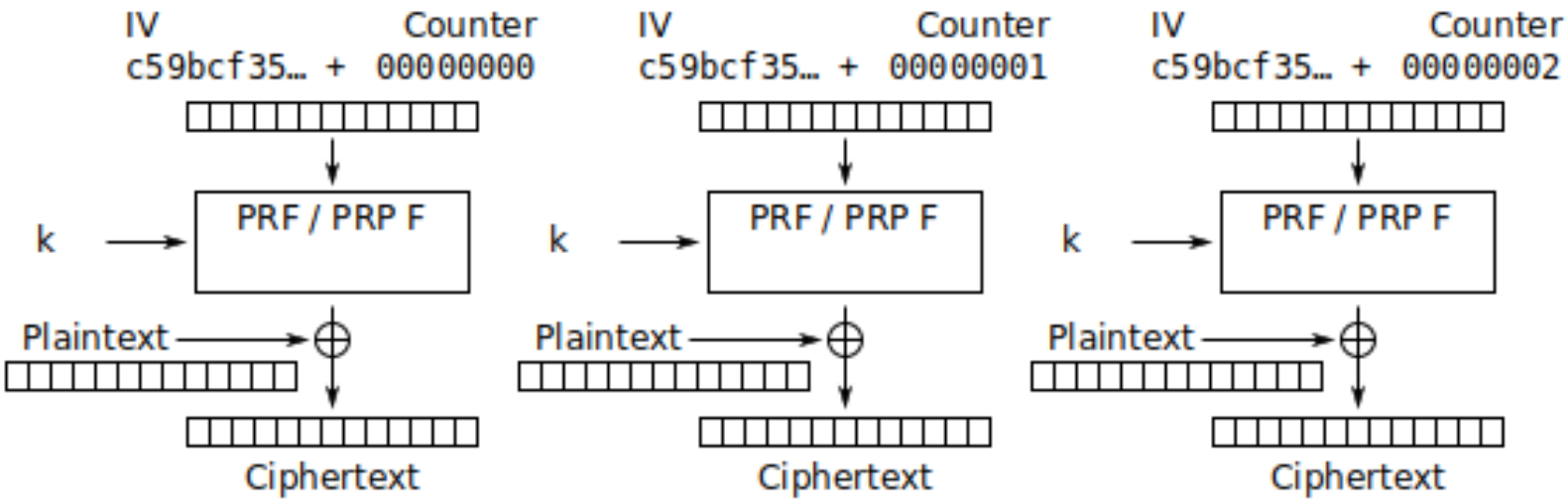
Cipher block chaining mode (CBC)



Cipher Block Chaining (CBC) mode encryption

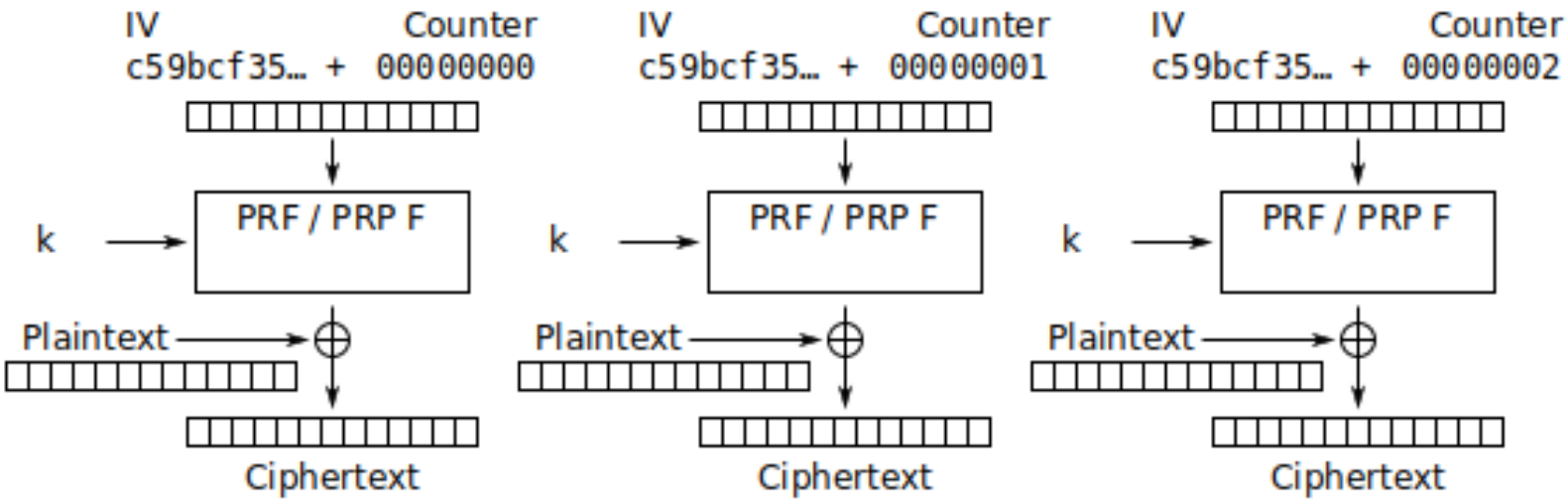
IND-CPA if F is a PRP. IV has to be random, if it is predictable CBC is vulnerable!

Counter mode (CTR)



Counter (CTR) mode encryption

Counter mode (CTR)



Counter (CTR) mode encryption

IND-CPA if F is a PRF.

What about active attacks?

- $\mathcal{A}$ might be able to learn decryption of ciphertexts at a later point by compromising the system.

What about active attacks?

- $\mathcal{A}$ might be able to learn decryption of ciphertexts at a later point by compromising the system.
- $\mathcal{A}$ might even get access to a decryption oracle (lunch time attack).

What about active attacks?

- $\mathcal{A}$ might be able to learn decryption of ciphertexts at a later point by compromising the system.
- $\mathcal{A}$ might even get access to a decryption oracle (lunch time attack).
- Want to model the worst case: Let $\mathcal{A}$ choose ciphertexts that get decrypted!

IND under chosen ciphertext attacks

$\mathrm{Exp}^{\mathrm{IND-CCA}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot),\mathrm{Dec}_k(\cdot)}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0 = m_1|$
3. $b \leftarrow_R \{0,1\}, c^* \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot),\mathrm{Dec}_k(\cdot)}(c^*)$ with $\mathrm{Dec}_k(c^*) = \perp$
5. Output 1 if $b' = b$, otherwise 0

IND under chosen ciphertext attacks

$\mathrm{Exp}^{\mathrm{IND-CCA}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot),\mathrm{Dec}_k(\cdot)}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0 = m_1|$
3. $b \leftarrow_R \{0,1\}, c^* \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot),\mathrm{Dec}_k(\cdot)}(c^*)$ with $\mathrm{Dec}_k(c^*) = \perp$
5. Output 1 if $b' = b$, otherwise 0

Definition (IND-CCA)

A secret key encryption scheme $\mathcal{E}$ has indistinguishable ciphertexts under chosen ciphertext attacks if for all efficient adversaries $\mathcal{A}$ their advantage ε in winning above game is negligible

$$\Pr\left[\mathrm{Exp}^{\mathrm{IND-CCA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] \leq \frac{1}{2} + \varepsilon.$$

IND under chosen ciphertext attacks

$\mathrm{Exp}^{\mathrm{IND-CCA}}_{\mathcal{E},\mathcal{A}}(n)$:

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $m_0, m_1 \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot),\mathrm{Dec}_k(\cdot)}(1^n)$ with $m_0, m_1 \in \mathcal{M} \wedge |m_0 = m_1|$
3. $b \leftarrow_R \{0,1\}, c^* \leftarrow \mathrm{Enc}_k(m_b)$
4. $b' \leftarrow \mathcal{A}^{\mathrm{Enc}_k(\cdot),\mathrm{Dec}_k(\cdot)}(c^*)$ with $\mathrm{Dec}_k(c^*) = \perp$
5. Output 1 if $b' = b$, otherwise 0

Definition (IND-CCA)

A secret key encryption scheme $\mathcal{E}$ has indistinguishable ciphertexts under chosen ciphertext attacks if for all efficient adversaries $\mathcal{A}$ their advantage ε in winning above game is negligible

$$\Pr\left[\mathrm{Exp}^{\mathrm{IND-CCA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] \leq \frac{1}{2} + \varepsilon.$$

This definition is equivalent to SEM-CCA.

MAC

Message authentication

- Sometimes we want more than secrecy!
- Acknowledgement of receipt, social communication, source of executable, ...

Message authentication

- Sometimes we want more than secrecy!
- Acknowledgement of receipt, social communication, source of executable, ...
- We need integrity and authenticity!

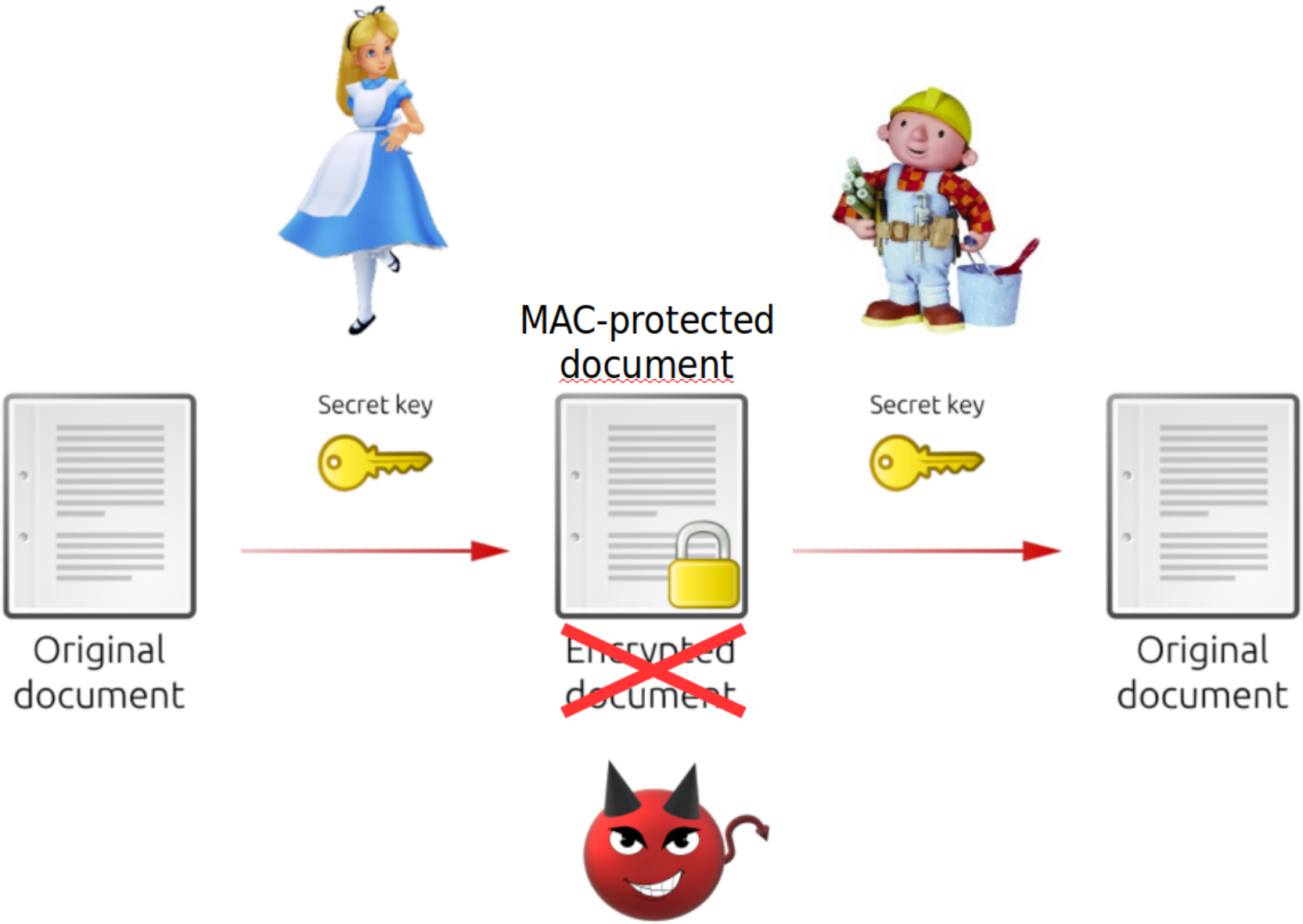
Message authentication

- Sometimes we want more than secrecy!
- Acknowledgement of receipt, social communication, source of executable, ...
- We need integrity and authenticity!
- Encryption $\stackrel{?}{\Rightarrow}$ Authenticity / integrity?

Message authentication

- Sometimes we want more than secrecy!
- Acknowledgement of receipt, social communication, source of executable, ...
- We need integrity and authenticity!
- Encryption $\stackrel{?}{\Rightarrow}$ Authenticity / integrity?
  - PRG-ENC, PRF-ENC, ... any stream cipher allows controlled bit-flips. If format is known this may be disastrous
  - Block ciphers make similar attacks harder but no guarantees.
  - ECB-mode allows to switch order of blocks, repeat blocks, etc.

MAC


MAC-protected document
Secret key
Secret key
Original document
Original document

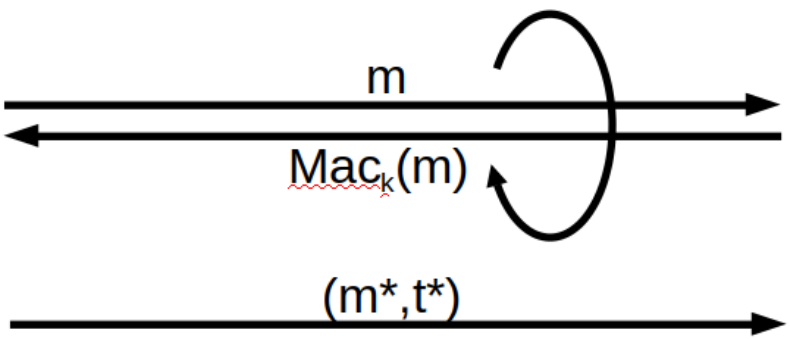
Message authentication codes (MAC)

Definition (message authentication code)

A message authentication code or MAC is a tuple of probabilistic polynomial-time algorithms $\text{MAC} = (\text{Gen}, \text{Mac}, \text{Vrfy})$ over a message space $\mathcal{M}$, fulfilling the following:

- Gen is a probabilistic algorithm that on input 1^n outputs a key k. The output space of Gen is called the key space $\mathcal{K}$.
- Mac takes as input a key $k \in \mathcal{K}$ and a message $m \in \mathcal{M}$, and outputs a tag $t \in \mathcal{T}$. The output space of Mac is called tag space $\mathcal{T}$.
- Vrfy is a deterministic algorithm that takes as inputs a key $k \in \mathcal{K}$, a message $m \in \mathcal{M}$, and a tag $t \in \mathcal{T}$, and outputs a bit $b \in \{0, 1\}$.
- Correctness: For every n, every $k \leftarrow \text{Gen}(1^n)$, and every $m \in \mathcal{M}$ it holds that $\text{Vrfy}_k(m, \text{Mac}_k(m)) = 1$.

Existential unforgeability under (adaptive) chosen message attacks (EU-CMA)



Existential unforgeability under (adaptive) chosen message attacks (EU-CMA)

$\mathsf{Exp}^{\mathrm{EU-CMA}}_{\mathsf{MAC},\mathcal{A}}(n)$

1. $k \leftarrow \mathrm{Gen}(1^n)$
2. $(m, t) \leftarrow \mathcal{A}^{\mathrm{MAC}_k(\cdot)}(1^n)$. Let $\{m_i\}_1^q$ denote $\mathcal{A}$'s queries to MAC_k
3. If $\mathrm{VRFY}_k(m, t) := 1$ and $m \notin \{m_i\}_1^q$ return 1
4. Else return 0.

Existential unforgeability under (adaptive) chosen message attacks (EU-CMA)

Definition (EU-CMA)

A message authentication code $\mathsf{MAC} = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$ over a message space $\mathcal{M}$ is existentially unforgeable under an adaptive chosen-message attack, or just secure, if for all efficient adversaries $\mathcal{A}$ the success probability ε in winning $\mathsf{Exp}^{\mathrm{EU-CMA}}_{\mathsf{MAC},\mathcal{A}}(n)$ is negligible, where

$$\varepsilon = \Pr\left[\mathsf{Exp}^{\mathrm{EU-CMA}}_{\mathsf{MAC},\mathcal{A}}(n) = 1\right]$$

Remarks

- There exists a constant time attack with success probability $1/|\mathcal{T}|$ against every MAC $\Rightarrow$ Tags must not be too short

Remarks

- There exists a constant time attack with success probability $1/|\mathcal{T}|$ against every MAC $\Rightarrow$ Tags must not be too short
- MAC's do not prevent replay attacks!
- Replay attacks have to be handled on protocol level (e.g., using sequence numbers).

PRF is a MAC

Theorem

A secure PRF F *leads a secure MAC with*

$\mathrm{Gen}(1^n)$ *returns* $k \leftarrow_R \{0,1\}^n$.

$\mathrm{MAC}_k(m)$ *returns* $t = \mathsf{F}_k(m)$.

$\mathrm{VRFY}_k(m,t)$ *returns 1 if* $t = \mathsf{F}_k(m)$*, and 0 otherwise.*

Proof idea

Build distinguisher that simulates experiment using its oracle instead of F. A valid forgery must be on a new message. So if oracle is random, tag is a correct guess for a random function at some point m that was not queried. If $\mathcal{A}$ succeeds more often when the oracle was F, this allows to distinguish F as for PRF-ENC.

CBC-MAC

Construction

Let F *be an efficient, length-preserving keyed function over* $\{0,1\}^n$. *CBC-MAC has message space* $\mathcal{M} = (\{0,1\}^{\ell n})$. *The algorithms are as follows:*

$\mathrm{Gen}(1^n)$ *returns* $k \leftarrow_R \{0,1\}^n$.

$\mathrm{MAC}_k(m)$ *upon input key* $k \in \{0,1\}^n$ *and a message* m *of length* ℓn, *do the following:*

1. *Denote* $m = m_1, \ldots, m_\ell$ *where each* m_i *is of length* n, *and set* $t_0 = 0^n$.
2. *For* $i = 1$ *to* ℓ, *set* $t_i \leftarrow \mathsf{F}_k(t_{i-1} \oplus m_i)$.
3. *Output* t_ℓ.

$\mathrm{VRFY}_k(m,t)$ *returns 1 if* $t = \mathrm{MAC}_k(m)$, *and 0 otherwise.*

Variable message length CBC-MAC

- CBC-MAC is not secure for variable length messages

Variable message length CBC-MAC

- CBC-MAC is not secure for variable length messages

Solutions for variable ℓ:

- Derived key: Compute $k' = \mathsf{F}_k(\ell)$ and use k' to compute $t = \mathrm{MAC}_{k'}(m)$
- Prepend length: Compute $t = \mathrm{MAC}_k(\ell \| m)$.
- Encrypted tag: Use two keys $k_1, k_2 \in \{0,1\}^n$, compute $t' = \mathrm{MAC}_{k_1}(m)$ and output $t = \mathsf{F}_{k_2}(t')$. We can generate k_1, k_2 from a single key using F as a length-doubling PRG ($< k_1, k_2 > = < \mathsf{F}_k(0), \mathsf{F}_k(1) >$)

Padding

- What if the message length is not a multiple of the block length: $|m| \neq x \cdot n$?

Padding

- What if the message length is not a multiple of the block length: $|m| \neq x \cdot n$?

Solution: Padding

- Expand message to match multiple of block length.
- Usually injective function $\mathrm{Pad} : \{0,1\}^* \to (\{0,1\}^n)^*$.
- E.g., $m \to m \| 10^*$.
- Properties depend on cryptographic application:
  - Encryption - invertible
  - MAC - injective
- Often used for additional purposes: Randomization, or encoding message length.

Secrecy + Authenticity

- We actually want IND-CCA and EU-CMA security of our connections.

Secrecy + Authenticity

- We actually want IND-CCA and EU-CMA security of our connections.

Options:

- Encrypt-and-MAC: $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(m)$.
- MAC-then-Encrypt. $t = \mathrm{MAC}_{k_2}(m),\ c = \mathrm{Enc}_{k_1}(m \| t)$.
- Encrypt-then-MAC. $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(c)$.

Secrecy + Authenticity

- We actually want IND-CCA and EU-CMA security of our connections.

Options:

- Encrypt-and-MAC: $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(m)$. Possibly insecure as MAC might leak!
- MAC-then-Encrypt. $t = \mathrm{MAC}_{k_2}(m),\ c = \mathrm{Enc}_{k_1}(m \| t)$.
- Encrypt-then-MAC. $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(c)$.

Secrecy + Authenticity

- We actually want IND-CCA and EU-CMA security of our connections.

Options:

- Encrypt-and-MAC: $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(m)$.
  Possibly insecure as MAC might leak!
- MAC-then-Encrypt. $t = \mathrm{MAC}_{k_2}(m),\ c = \mathrm{Enc}_{k_1}(m \| t)$.
  Possibly insecure but counter-examples are more involved
- Encrypt-then-MAC. $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(c)$.

Secrecy + Authenticity

- We actually want IND-CCA and EU-CMA security of our connections.

Options:

- Encrypt-and-MAC: $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(m)$.
  Possibly insecure as MAC might leak!
- MAC-then-Encrypt. $t = \mathrm{MAC}_{k_2}(m),\ c = \mathrm{Enc}_{k_1}(m \| t)$.
  Possibly insecure but counter-examples are more involved
- Encrypt-then-MAC. $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(c)$.
  Secure! (And the generic way to turn an IND-CPA secure encryption into an IND-CCA secure one.)

Secrecy + Authenticity

- We actually want IND-CCA and EU-CMA security of our connections.

Options:

- Encrypt-and-MAC: $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(m)$.
  Possibly insecure as MAC might leak!
- MAC-then-Encrypt. $t = \mathrm{MAC}_{k_2}(m),\ c = \mathrm{Enc}_{k_1}(m \| t)$.
  Possibly insecure but counter-examples are more involved
- Encrypt-then-MAC. $c = \mathrm{Enc}_{k_1}(m),\ t = \mathrm{MAC}_{k_2}(c)$.
  Secure! (And the generic way to turn an IND-CPA secure encryption into an IND-CCA secure one.)
- Or the most simple one: Use Authenticated encryption (AE)!

Conclusion

- We covered secret key encryption schemes and their security.
- We covered message authentication codes and their security.
- On the way we looked at PRFs and PRGs.

Conclusion

- We covered secret key encryption schemes and their security.
- We covered message authentication codes and their security.
- On the way we looked at PRFs and PRGs.

Thank you!
Questions?