# Cryptography Part IV: Encryption Modes

## *ECE 4156/6156 Hardware-Oriented Security and Trust*

Spring 2024

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

# Reading Assignment

- Please read Chapter 3 of the course textbook by Katz and Lindell
- Please read Chapter 2 of the course textbook by Menezes, Oorschot and Vanstone, i.e., the Handbook of Applied Cryptography
  - Note: this book will be referred to later in these notes as "HAC"

# Notation from HAC (pages 49 and 50)

- $\mathbb{R}$ is the set of real numbers, e.g., $\pi \in \mathbb{R}$ while $\sqrt{-1} \notin \mathbb{R}$
- $\mathbb{Z}$ is the set of integers, i.e., $\mathbb{Z}$ = {…,-3,-2,-1,0,1,2,3,…}
- $f: A \rightarrow B$ is a function that maps each $a \in A$ to precisely one $b \in B$. Given that $f(a) = b$, then $b$ is called the *image* of $a$, and $a$ is called the *preimage* of $b$. The set $A$ is called the *domain* of $f$.
- A function $f: A \rightarrow B$ is 1 – 1 (*one-to-one*) or *injective* if each element in $B$ is the image of at most one element in $A$. Hence $f(a_1) = f(a_2)$ implies $a_1 = a_2$.
- A function $f: A \rightarrow B$ is *onto* or *surjective* if each $b \in B$ is the image of at least one $a \in A$.
- A function $f: A \rightarrow B$ is a *bijection* if it is both one-to-one and onto. If $f$ is a bijection between finite sets $A$ and $B$, then $|A| = |B|$. If $f$ is a bijection between a set $A$ and itself, then $f$ is called a *permutation* on $A$.

# Additional Notation (from Prof. Mooney)

- $\mathbb{N}$ is the set of natural numbers, i.e., $\mathbb{N}$ = {1,2,3,…}
- $f: A \to B$ is a function that maps each $a \in A$ to precisely one $b \in B$. Given that $f(a) = b$, then $b$ is called the *image* of $a$, and $a$ is called the *preimage* of $b$. The set $A$ is called the *domain* of $f$. The set $B$ is called the *range* of $f$.

# Notation from Katz and Lindell

- $\{X\}$ is a set of elements of type $X$
- $m$ is a message in plaintext
  - $m$ is composed of smaller blocks $m_i$ suitable for individual encryption steps
  - $m = \{m_i\}$
- $c_i$ is ciphertext corresponding to message block $m_i$
- $c$ is ciphertext corresponding to message $m$
- $Enc_k$ is encryption with key $k$
  - $c \leftarrow Enc_k(m)$ (NOTE: there may be multiple valid ciphertexts!!!)
  - $c := Enc_k(m)$ (NOTE: deterministic, i.e., there is only one valid ciphertext)
- $Dec_k$ is decryption with key $k$
  - $m := Dec_k(c)$ (NOTE: deterministic, i.e., there is only one valid message)
- $<a,b>$ is a concatenation of $a$ followed by $b$
- $a||b$ is unambiguous concatenation of $a$ followed by $b$; “unambiguous concatenation” means that $a$ and $b$ can be recovered from $a||b$

# Notation from Katz and Lindell (continued)

- $\mathrm{PrivK}$ is an experiment involving a private key
- $A$ is an adversary
- $\mathrm{eav}$ refers to eavesdropping and obtaining ciphertext only
- $\pi$ = $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is an encryption scheme
- $\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}$ is an experiment involving a private key encryption scheme $\pi$ with an adversary $A$ only with access to ciphertext
- $\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}(n)$ is an experiment involving a private key encryption scheme $\pi$ with a key of size $n$ and an adversary $A$ only with access to ciphertext
- $\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}(n,0)$ is an experiment involving a private key encryption scheme $\pi$ with a key of size $n$, message selection bit $b$=0 and an adversary $A$ only with ciphertext[1]
- $A$ does not have access to additional information, e.g., $A$ does not have valid plaintext-ciphertext pairs obtained through other means
- $\mathrm{Probabilistic\ Polynomial\ Time}$ or $\mathrm{PPT}$ refers to algorithms which take at most polynomial time while having free use of a true random number generator

[1] Page 55 of Katz and Lindell.

# Recall Slide 11 from Crypto I Lecture

- $M$ is a set of all possible messages, i.e., the message space
- $C$ is a set of all possible ciphertexts, i.e., the ciphertext space
- *Gen* is a key generation procedure
  - The output of *Gen* is key $k$
  - *Gen* may or may not require an input

# Now We Add the Following

- $K$ is a set of all possible keys, i.e., the key space
- In the one-time pad, $|K| = |M| = |C| = \ell$

# Where We Are So Far: Status

***DEFINITION 2.5*** *Encryption scheme* $\pi$ = (Gen, Enc, Dec) *with message space* $M$ *is* **perfectly indistinguishable** *if for every* $A$ *it holds that*

$$\Pr\left[\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}} = 1\right] = \frac{1}{2}.$$

# Where We Are So Far: Status

***DEFINITION 2.5*** *Encryption scheme* $\pi$ = (Gen, Enc, Dec) *with message space* $M$ *is* **perfectly indistinguishable** *if for every* $A$ *it holds that*

$$\Pr\left[\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}} = 1\right] = \frac{1}{2}.$$

***DEFINITION 3.8*** *A private-key encryption scheme* $\pi$ = (Gen, Enc, Dec) *has* **indistinguishable encryptions in the presence of an eavesdropper***, or is* **EAV-secure***, if for all* PPT *adversaries* $A$ *there is a negligible function* negl *such that, for all n,*

$$\Pr\left[\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}(n) = 1\right] \leq \frac{1}{2} + \mathrm{negl}(n),$$

*where the probability is taken over the randomness used by* $A$ *and the randomness used in the experiment (for choosing the key and bit b, as well as any randomness used by* Enc*).*

# Where We Are So Far: Status

***DEFINITION 3.8*** *A private-key encryption scheme* $\pi$ = (Gen, Enc, Dec) *has* **indistinguishable encryptions in the presence of an eavesdropper***, or is* **EAV-secure***, if for all* PPT *adversaries* $A$ *there is a negligible function* negl *such that, for all n,*

$$\Pr\left[\mathrm{PrivK}^{\mathrm{eav}}_{A,\pi}(n) = 1\right] \leq \frac{1}{2} + \mathrm{negl}(n),$$

*where the probability is taken over the randomness used by* $A$ *and the randomness used in the experiment (for choosing the key and bit b, as well as any randomness used by* Enc*).*

# Where We Are So Far: Status (continued)

***DEFINITION 3.9*** *A private-key encryption scheme* $\pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *has* **indistinguishable encryptions in the presence of an eavesdropper** *if for all* PPT *adversaries* $A$ *there is a negligible function* negl *such that*

$$\left|\Pr\left[\mathrm{out}_A(\mathrm{PrivK}^{\mathrm{eav}}_{A,\pi}(n,0)) = 1\right] - \Pr\left[\mathrm{out}_A(\mathrm{PrivK}^{\mathrm{eav}}_{A,\pi}(n,1)) = 1\right]\right| \leq \mathrm{negl}(n).$$

# Where We Are So Far: Status (continued)

***DEFINITION 3.9*** *A private-key encryption scheme* $\pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *has* **indistinguishable encryptions in the presence of an eavesdropper** *if for all* PPT *adversaries* $A$ *there is a negligible function* negl *such that*

$$\left|\Pr\left[\mathrm{out}_A(\mathrm{PrivK}^{\mathrm{eav}}_{A,\pi}(n,0)) = 1\right] - \Pr\left[\mathrm{out}_A(\mathrm{PrivK}^{\mathrm{eav}}_{A,\pi}(n,1)) = 1\right]\right| \leq \mathrm{negl}(n).$$

***THEOREM 3.10*** *Let* $\pi = (\mathrm{Enc}, \mathrm{Dec})$ *be a fixed-length private-key encryption scheme for messages of length* $\ell$ *that has indistinguishable encryptions in the presence of an eavesdropper. Then for all* PPT *adversaries* $A$ *and any* $i \in \{1,\ldots,\ell\}$, *there is a negligible function* negl *such that*

$$\Pr\left[A(1^n, \mathrm{Enc}_k(m)) = m^i\right] \leq \frac{1}{2} + \mathrm{negl}(n),$$

*where the probability is taken over uniform* $m \in \{0,1\}^\ell$ *and* $k \in \{0,1\}^n$, *the randomness of* $A$, *and the randomness of* Enc.

# Where We Are So Far: Status (continued)

***THEOREM 3.10*** *Let* $\pi$ = $(\mathrm{Enc}, \mathrm{Dec})$ *be a fixed-length private-key encryption scheme for messages of length* $\ell$ *that has indistinguishable encryptions in the presence of an eavesdropper. Then for all* $\mathrm{PPT}$ *adversaries* $A$ *and any* $i \in \{1,\ldots,\ell\}$*, there is a negligible function* $\mathrm{negl}$ *such that*

$$\Pr\left[A(1^n, \mathrm{Enc}_k(m)) = m^i\right] \leq \frac{1}{2} + \mathrm{negl}(n),$$

*where the probability is taken over uniform* $m \in \{0,1\}^\ell$ *and* $k \in \{0,1\}^n$*, the randomness of* $A$*, and the randomness of* $\mathrm{Enc}$*.*

**DEFINITION 3.14** *Let $\ell$ be a polynomial and let $G$ be a deterministic polynomial-time algorithm such that for any $n$ and any input $s \in \{0,1\}^n$, the result $G(s)$ is a string of length $\ell(n)$. We say that $G$ is a* pseudorandom generator *if the following conditions hold:*

1. **(Expansion:)** *For every $n$ it holds that $\ell(n) > n$.*
2. **(Pseudorandomness:)** *For any* PPT *algorithm $D$, there is a negligible function* negl *such that*

$$\left| \Pr[D(G(s)) = 1] - \Pr[D(r) = 1] \right| \leq \mathsf{negl}(n),$$

*where the first probability is taken over uniform choice of $s \in \{0,1\}^n$ and the randomness of $D$, and the second probability is taken over uniform choice of $r \in \{0,1\}^{\ell(n)}$ and the randomness of $D$.*

*We call $\ell$ the* expansion factor *of $G$.*

***ALGORITHM 3.16***

**Constructing $G_\ell$ from** $(\mathsf{Init}, \mathsf{GetBits})$

**Input:** Seed $s$ and optional initialization vector $IV$

**Output:** $y_1, \ldots, y_\ell$

$\mathsf{st}_0 := \mathsf{Init}(s, IV)$

**for** $i = 1$ to $\ell$:

  $(y_i, \mathsf{st}_i) := \mathsf{GetBits}(\mathsf{st}_{i-1})$

**return** $y_1, \ldots, y_\ell$

***CONSTRUCTION 3.17***

Let $G$ be a pseudorandom generator with expansion factor $\ell$. Define a private-key encryption scheme for messages of length $\ell$ as follows:

- Gen: on input $1^n$, choose uniform $k \in \{0,1\}^n$ and output it as the key.
- Enc: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^{\ell(n)}$, output the ciphertext
$$c := G(k) \oplus m.$$
- Dec: on input a key $k \in \{0,1\}^n$ and a ciphertext $c \in \{0,1\}^{\ell(n)}$, output the message
$$m := G(k) \oplus c.$$

A private-key encryption scheme based on any pseudorandom generator.

**THEOREM 3.18** *If $G$ is a pseudorandom generator, then Construction 3.17 is a fixed-length private-key encryption scheme that has indistinguishable encryptions in the presence of an eavesdropper.*

**PROOF** Let $\Pi$ denote Construction 3.17. We show that $\Pi$ satisfies Definition 3.8. Namely, we show that for any probabilistic polynomial-time adversary $\mathcal{A}$ there is a negligible function $\mathsf{negl}$ such that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n). \tag{3.2}$$

# Result(s)

- Given a pseudorandom number generator (PRNG) *G*
  - An exact example has yet to be provided
  - Definition 3.14, however, provides a framework to evaluate pseudorandom number generators
  - A PRNG efficiently expands a uniform (random) seed into a much larger pseudorandom output
    - Keeping the output length under a specified length provides number sequences which have no currently known way to be efficiently distinguished from a truly random number sequence
    - After the length is reached, use a new seed; note also the seed should be large, e.g., 128 bits, so than an adversary cannot guess the seed with any non-negligible probability of success
    - The seeds should be generated by a truly random physical process
  - No formal proof that PRNG's exist has been provided; but many practical constructions exist
- Construction 3.17 defines an encryption scheme $\pi$ using *G*
- Theorem 3.18 proves that Construction 3.17 is $\mathrm{EAV}$-secure

box" that encrypts messages of $\mathcal{A}$'s choice using a key $k$ that is unknown to $\mathcal{A}$. That is, we imagine $\mathcal{A}$ has access to an "oracle" $\mathsf{Enc}_k(\cdot)$; when $\mathcal{A}$ *queries* this oracle by providing it with a message $m$ as input, the oracle returns a ciphertext $c \leftarrow \mathsf{Enc}_k(m)$ as the reply. (When $\mathsf{Enc}$ is randomized, the oracle uses fresh randomness each time it answers a query.) The adversary is allowed to interact with the encryption oracle adaptively, as many times as it likes.

Consider the following experiment defined for any encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, adversary $\mathcal{A}$, and value $n$ for the security parameter:

**The CPA indistinguishability experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$:**

1. *A key $k$ is generated by running $\mathsf{Gen}(1^n)$.*
2. *The adversary $\mathcal{A}$ is given input $1^n$ and oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a pair of messages $m_0, m_1$ of the same length.*
3. *A uniform bit $b \in \{0,1\}$ is chosen, and then a ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$.*
4. *The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a bit $b'$.*
5. *The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. In the former case, we say that $\mathcal{A}$* succeeds.

**DEFINITION 3.22** *A private-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable encryptions under a chosen-plaintext attack, *or is* CPA-secure, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$ *there is a negligible function* negl *such that*

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n),$$

*where the probability is taken over the randomness used by* $\mathcal{A}$*, as well as the randomness used in the experiment.*

# This Concludes Where We Are So Far!!!

# Construction 3.17 is not CPA-secure

- Why?

# Construction 3.17 is not CPA-secure

- Why?
- In the CPA indistinguishability experiment $\mathrm{PrivK}_{A,\pi}^{\mathrm{cpa}}(n)$ step 2 provides oracle access to $\mathrm{Enc}_k(\cdot)$
  - (see page 74 of Katz and Lindell for the full list of steps)
  - Note that even though key k is secret, the adversary nonetheless has access to $\mathrm{Enc}_k(\cdot)$
- In step 4 the adversary continues to have oracle access prior to issuing a decision
- Clearly the adversary can simply compute $\mathrm{Enc}_k(m_0)$ and $\mathrm{Enc}_k(m_1)$!

# Keyed Functions[2]

- A keyed function $F: \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ has two inputs where the first is the key *k*
- Typically the inputs and output all have the same size *n*
  - Given key *k*, the keyed function is $F_k$
  - Then we have $F_k: \{0,1\}^n \rightarrow \{0,1\}^n$ where $F_k(x) = F(k, x)$

[2] Page 77 of Katz and Lindell.

# Pseudorandom Functions

- Keyed function $F_k$ is a **pseudorandom function** if for all $\mathrm{PPT}$ distinguishers $\mathrm{D}$ the chance that $\mathrm{D}$ can distinguish $F_k$ is from a uniform function $f$ is negligible.[3]
  - Note that a uniform function is not necessarily bijective
    - If $F_k: \{0,1\}^n \rightarrow \{0,1\}^n$, the comparable uniform function $f: \{0,1\}^n \rightarrow \{0,1\}^n$ may possibly have $f(x) = f(y)$ for $x \neq y$ with probability $\frac{1}{2^n}$

[3] See Def. 3.25 on page 79 of Katz and Lindell.

# Pseudorandom Permutation

- Keyed function $F_k$ is a **pseudorandom permutation** if for all $\mathrm{PPT}$ distinguishers $\mathrm{D}$ the chance that $\mathrm{D}$ can distinguish $F_k$ is from a uniform permutation $f$ is negligible.[4]
  - Function $f$:$\{0,1\}^n \rightarrow \{0,1\}^n$ is a uniform permutation if it is bijective.

- In practice, for sufficiently large n, the distinction between a uniform function and a uniform permutation is indistinguishable.[4]

[4] Page 80 of Katz and Lindell.

- A uniform function $f: \{0,1\}^n \rightarrow \{0,1\}^n$ is deterministic, i.e., for each input the output is defined, known and does not change
- The inverse of a uniform function $f: \{0,1\}^n \rightarrow \{0,1\}^n$, i.e., $f^{-1}: \{0,1\}^n \rightarrow \{0,1\}^n$ is typically not going to be deterministic because there may be an input with multiple valid outputs
- The inverse of a uniform function $f: A \rightarrow B$, i.e., $f^{-1}: B \rightarrow A$ is typically not going to be deterministic because there may be an input with multiple valid outputs

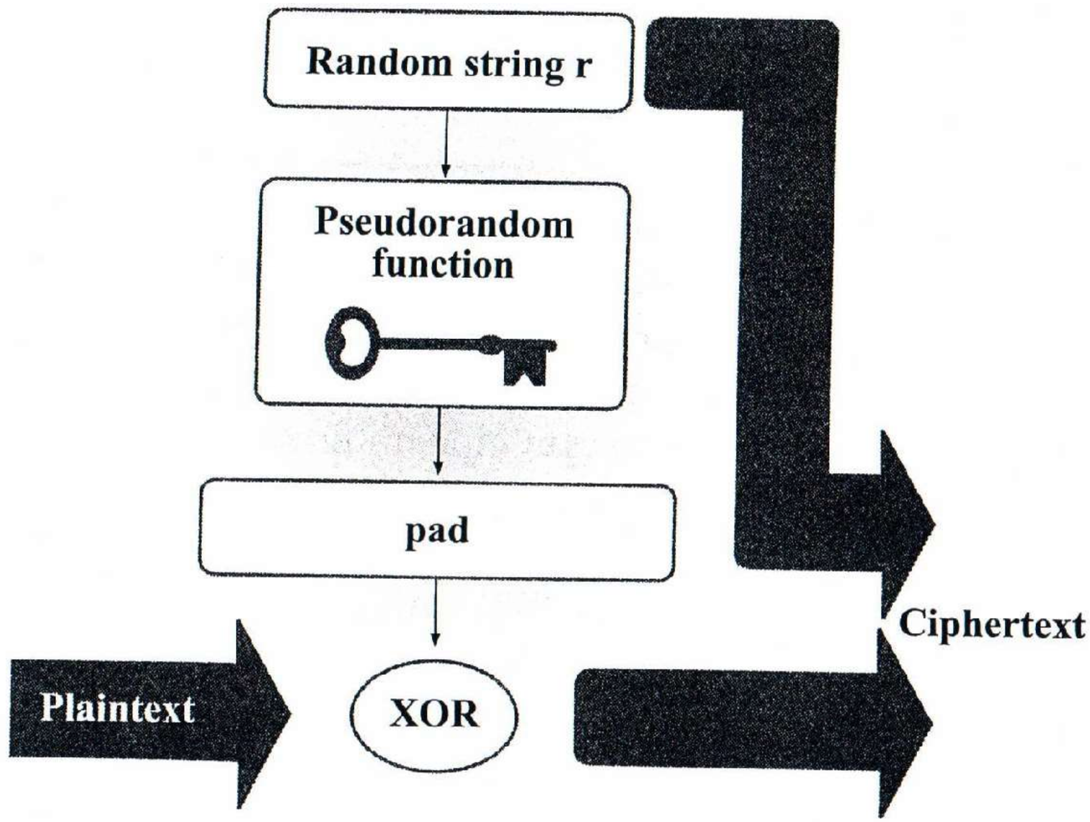



FIGURE 3.3: Encryption with a pseudorandom function.

***CONSTRUCTION 3.29***

Let $F$ be a pseudorandom function. Define a stream cipher (Init, GetBits), where each call to GetBits outputs $n$ bits, as follows:

- Init: on input $s \in \{0,1\}^n$ and $IV \in \{0,1\}^n$, set $\mathsf{st}_0 := (s, IV)$.
- GetBits: on input $\mathsf{st}_i = (s, IV)$, compute $IV' := IV + 1$ and set $y := F_s(IV')$ and $\mathsf{st}_{i+1} := (s, IV')$. Output $(y, \mathsf{st}_{i+1})$.

A stream cipher from any pseudorandom function/block cipher.

**CONSTRUCTION 3.30**

Let $F$ be a pseudorandom function. Define a private-key encryption scheme for messages of length $n$ as follows:

- Gen: on input $1^n$, choose uniform $k \in \{0,1\}^n$ and output it.
- Enc: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, choose uniform $r \in \{0,1\}^n$ and output the ciphertext

$$c := \langle r,\ F_k(r) \oplus m \rangle.$$

- Dec: on input a key $k \in \{0,1\}^n$ and a ciphertext $c = \langle r, s \rangle$, output the plaintext message

$$m := F_k(r) \oplus s.$$

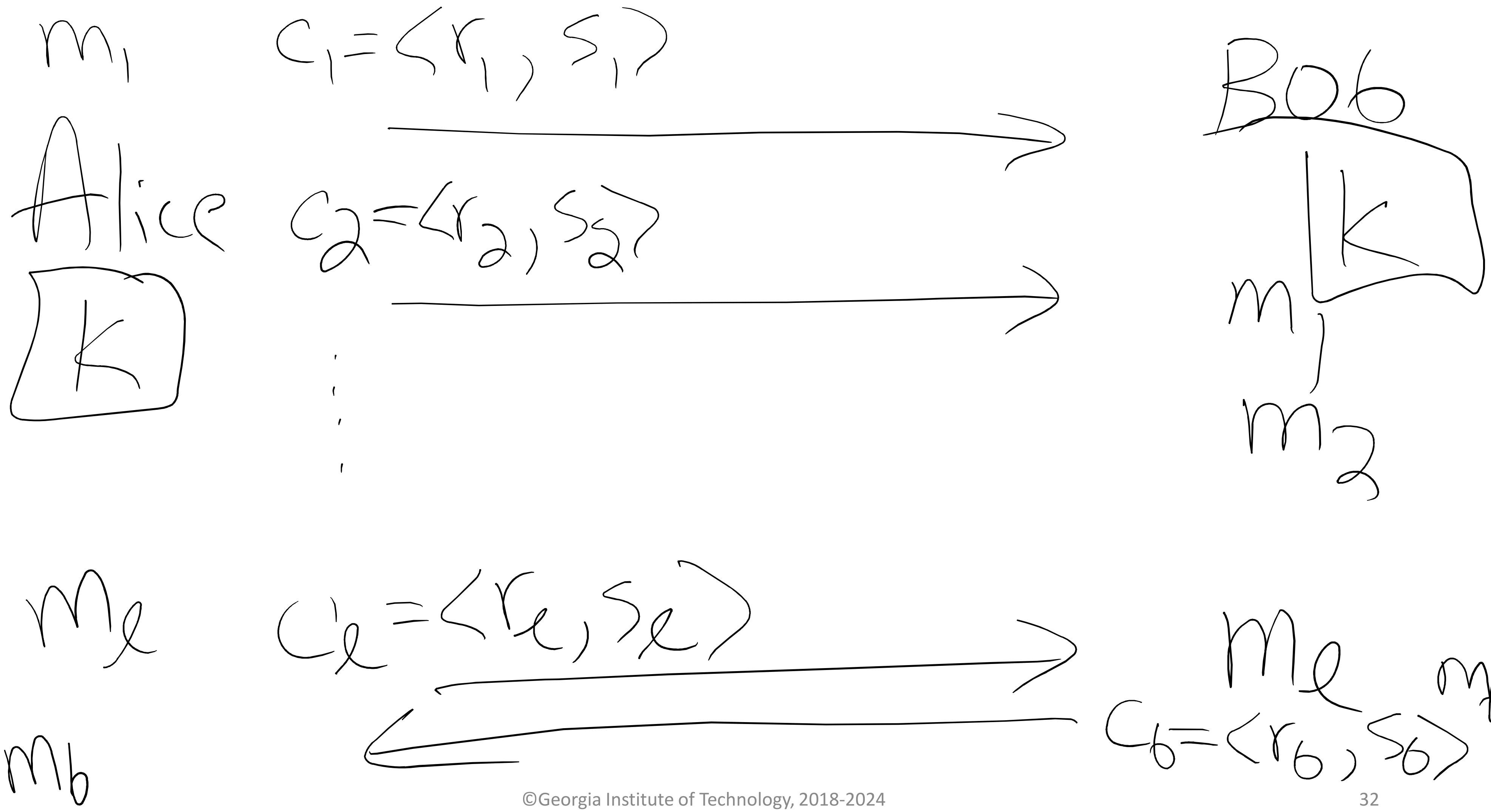
A CPA-secure encryption scheme from any pseudorandom function.

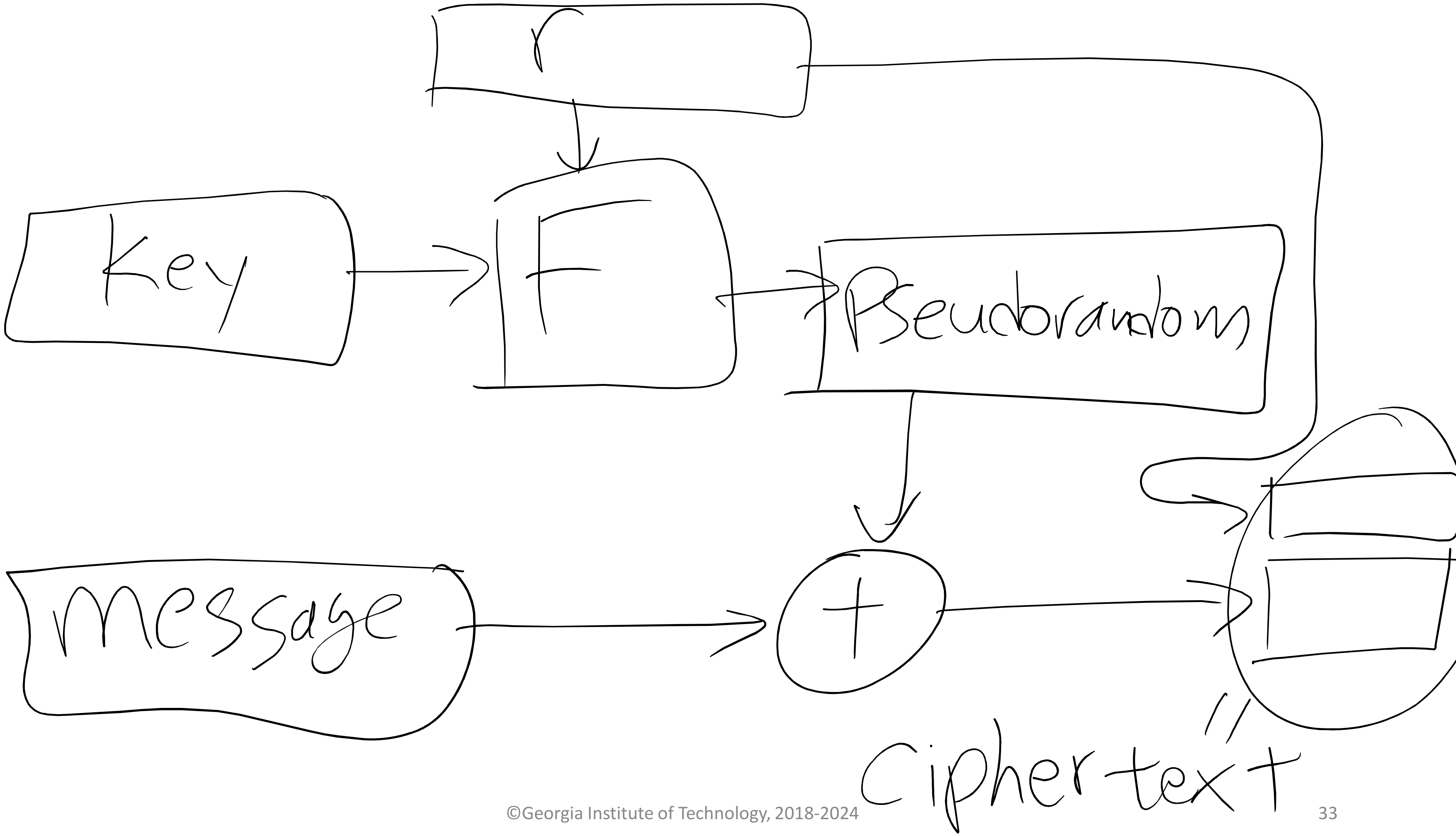
**CONSTRUCTION 3.30**

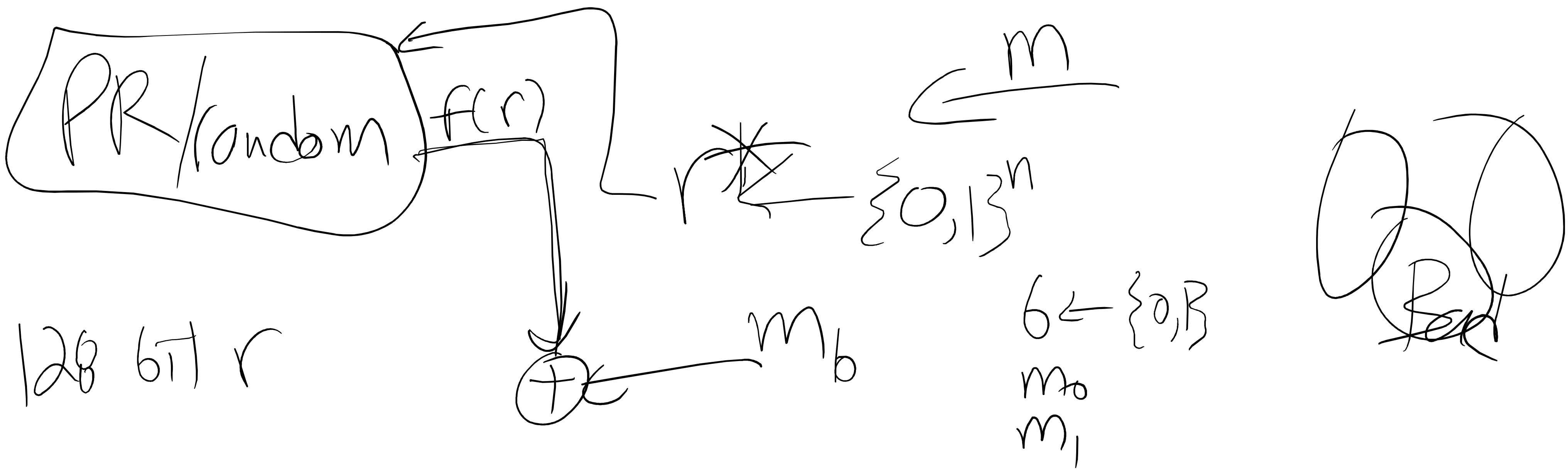
Let $F$ be a pseudorandom function. Define a private-key encryption scheme for messages of length $n$ as follows:

- Gen: on input $1^n$, choose uniform $k \in \{0,1\}^n$ and output it.
- Enc: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, choose uniform $r \in \{0,1\}^n$ and output the ciphertext
$$c := \langle r,\, F_k(r) \oplus m \rangle.$$
- Dec: on input a key $k \in \{0,1\}^n$ and a ciphertext $c = \langle r, s \rangle$, output the plaintext message
$$m := F_k(r) \oplus s.$$

A CPA-secure encryption scheme from any pseudorandom function.


m1
c1=<r1, s1>
Alice
c2=<r2, s2>
K
Bob
K
m1
m2
me
ce=<re, se>
me
mb
mb
cb=<rb, sb>


r
Key
F
Pseudorandom
message
+
ciphertext



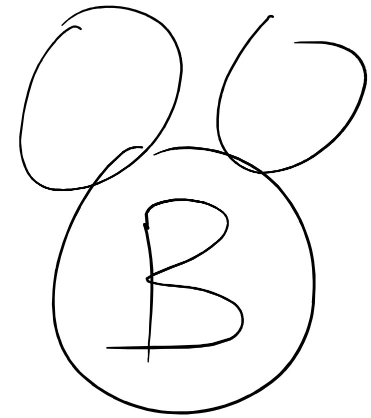

$r^*, f(r^*) \oplus m$ → $negl = \frac{polynomial}{2^{128}}$

$r^*$ = case that $r^*$ chosen = ciphertext random

||

$\left(f_K(r_a) \oplus \begin{matrix} m_0 \\ m_1 \end{matrix}\right) = S_a$ ?

$f_{K_{guess}}(r_a)$




$m_0, m_1,$

Ciphertext

$\langle r_a, S_a \rangle$

# Given F is Pseudorandom, Construction 3.30 is CPA-secure

- I hereby state the following:
- "The book goes through the proof in more detail, I just want you to get the intuition behind why Construction 3.30 is CPA-secure…I am not going to assign the proof on a homework or a test, guaranteed, …, however, **understanding** the intuition behind the proof is required and could be asked on a homework or a test!"

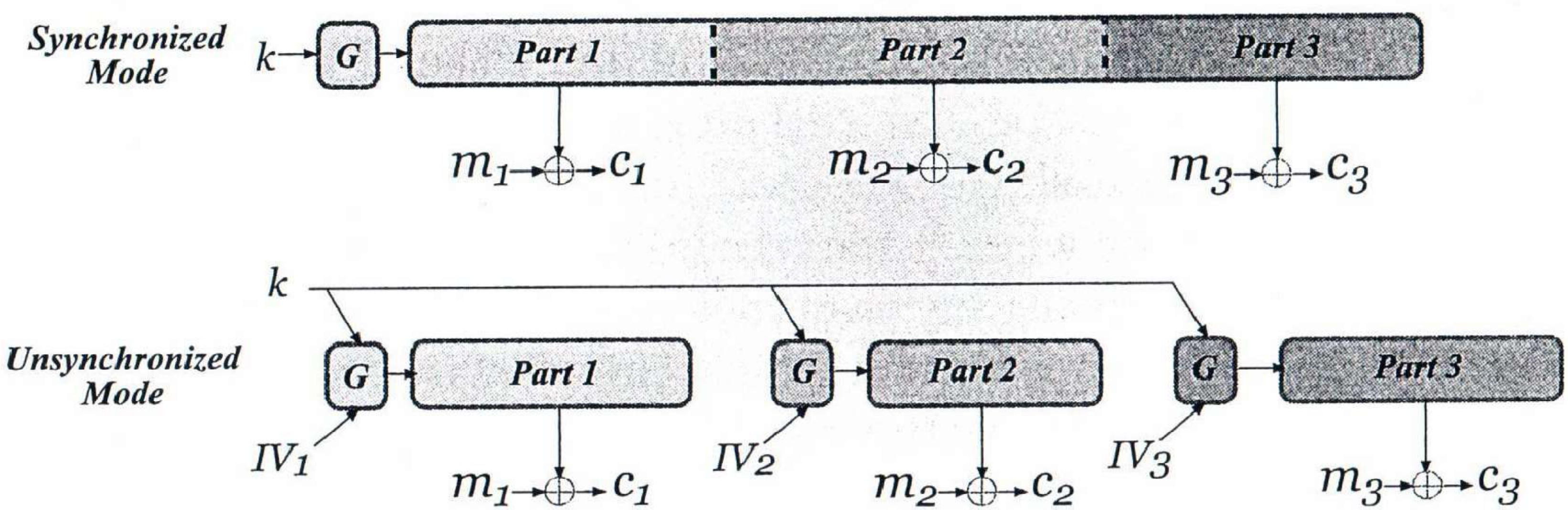



**FIGURE 3.4**: Synchronized mode and unsynchronized mode.

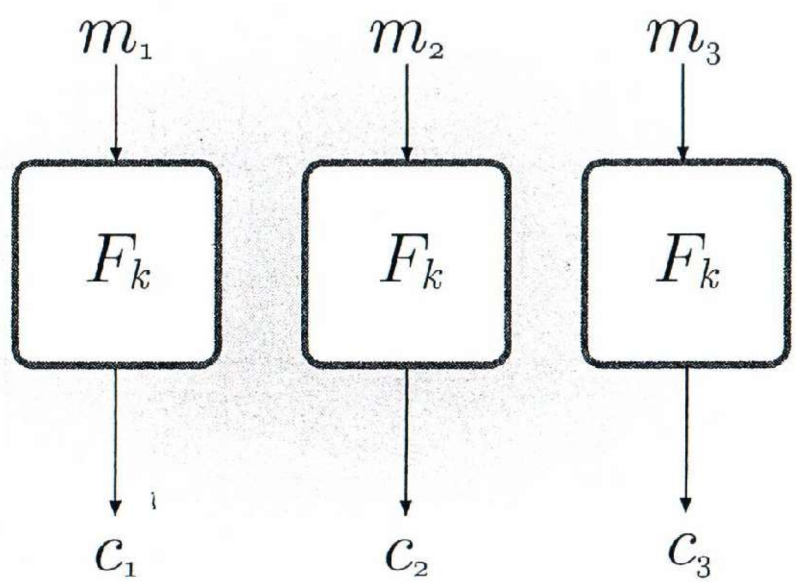



**FIGURE 3.5**: Electronic Code Book (ECB) mode.

Figure 3.5. Decryption is done in the obvious way, using the fact that $F_k^{-1}$ is efficiently computable.

$c_1 = Enc_k(m_1)$

$c_2 = Enc_k(m_2)$

⋮

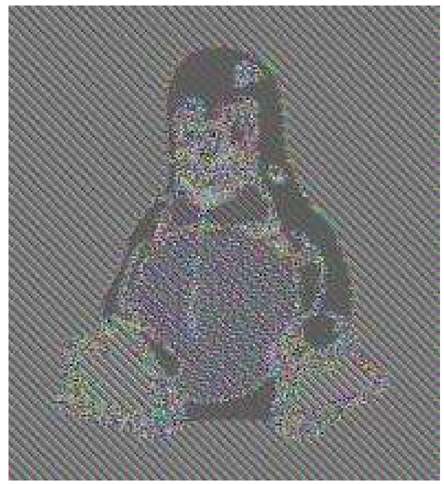
Problem: Deterministic?

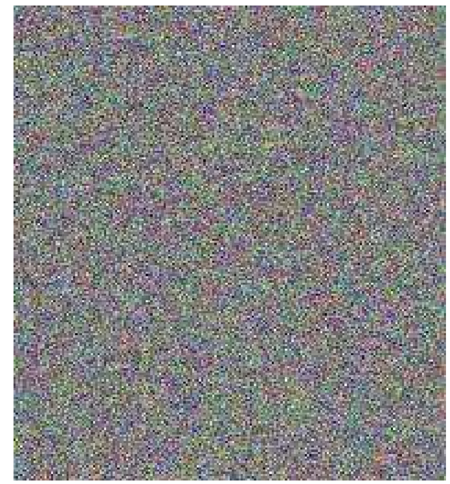
if $c_i = c_j \Rightarrow m_i = m_j$

not provide indist. enc.

For these reasons, ECB mode should never be used. (We include it only because of its historical significance.)

**FIGURE 3.6**: An illustration of the dangers of using ECB mode. The middle figure is an encryption of the image on the left using ECB mode; the figure on the right is an encryption of the same image using a secure mode. (Taken from `http://en.wikipedia.org` and derived from images created by Larry Ewing (`lewing@isc.tamu.edu`) using The GIMP.)





[2] From https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation and available under an open source license from Creative Commons.

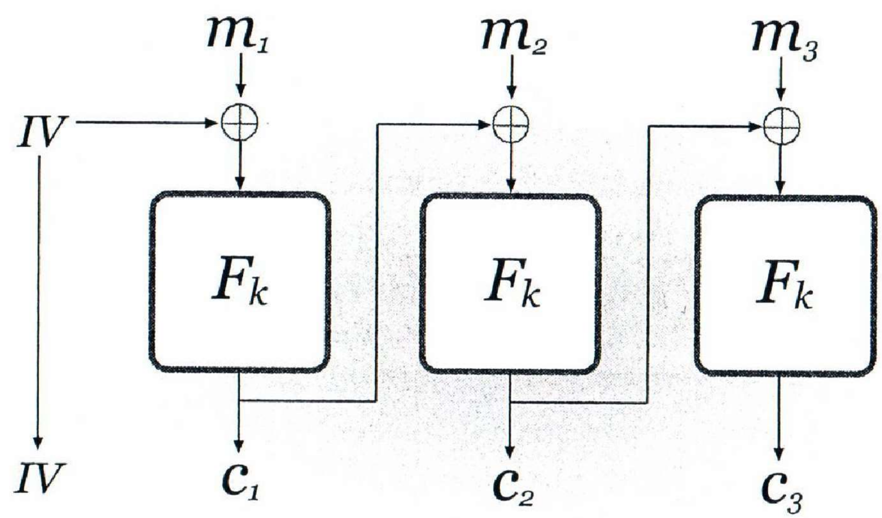



**FIGURE 3.7**: Cipher Block Chaining (CBC) mode.

**Cipher Block Chaining (CBC) mode.** To encrypt using this mode, a uniform initialization vector ($IV$) of length $n$ is first chosen. Then, ciphertext blocks are generated by applying the block cipher to the XOR of the current plaintext block and the previous ciphertext block. That is, set $c_0 := IV$ and then, for $i = 1$ to $\ell$, set $c_i := F_k(c_{i-1} \oplus m_i)$. The final ciphertext is $\langle c_0, c_1, \ldots, c_\ell \rangle$. (See Figure 3.7.) Decryption of a ciphertext $c_0, \ldots, c_\ell$ is done

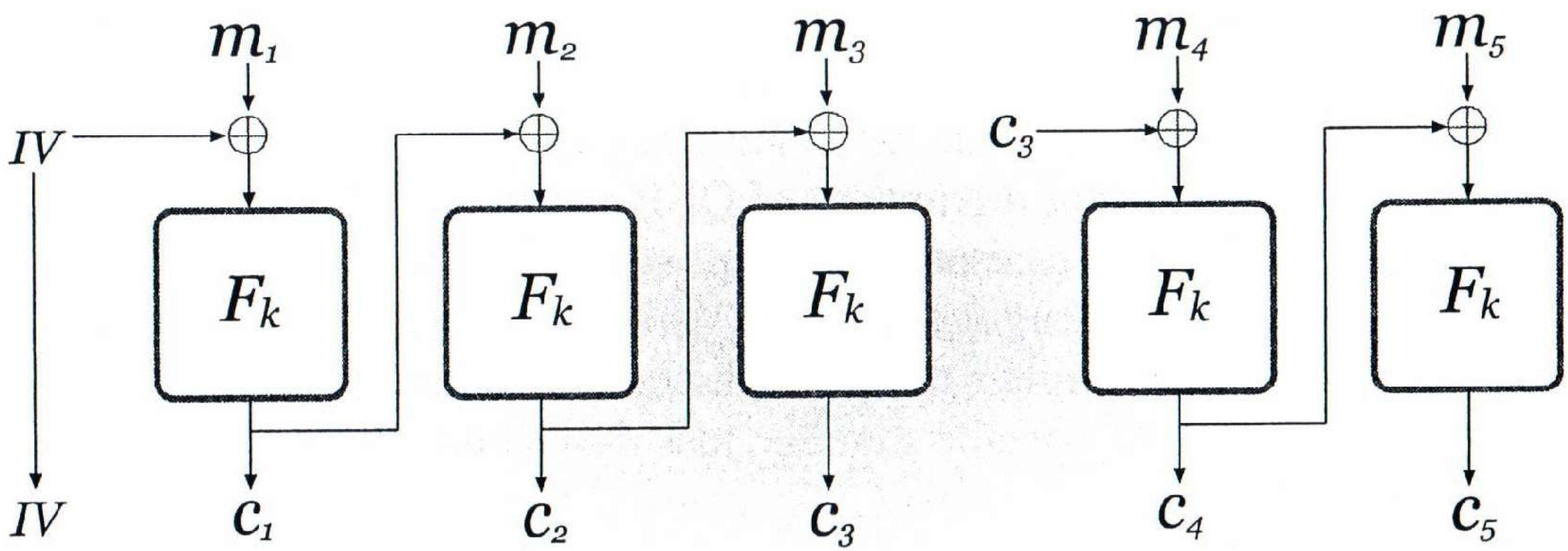



**FIGURE 3.8**: Chained CBC.

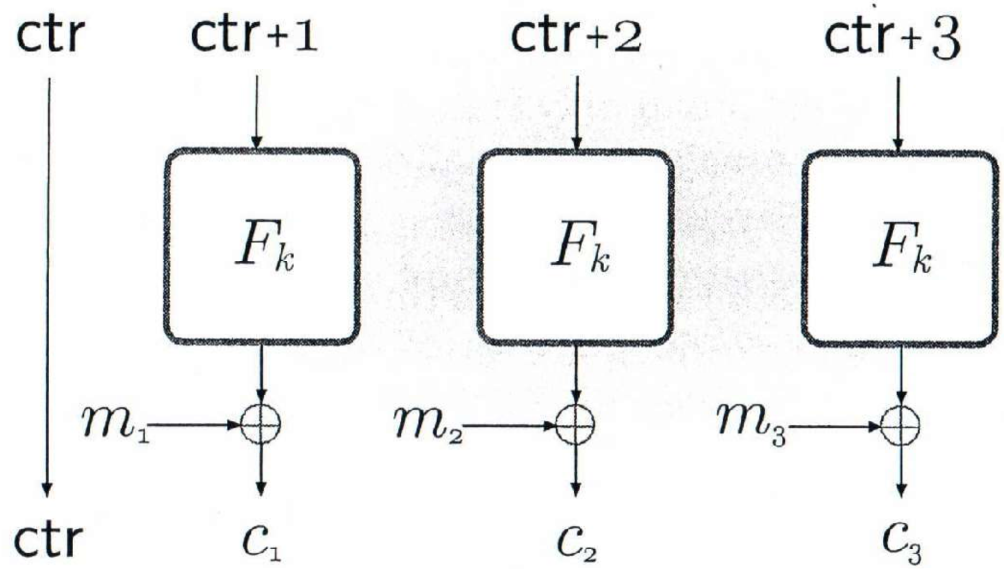



**FIGURE 3.10**: Counter (CTR) mode.

**Counter (CTR) mode.** Counter mode can also be viewed as an unsynchronized stream-cipher mode, where the stream cipher is constructed from the block cipher as in Construction 3.29. We give a self-contained description here. To encrypt using CTR mode, a uniform value $\mathsf{ctr} \in \{0,1\}^n$ is first chosen. Then, a pseudorandom stream is generated by computing $y_i := F_k(\mathsf{ctr} + i)$,

# Multiple Encryptions

- Ch. 3.4 of Katz and Lindell defines a multiple-message eavesdropping experiment $\mathrm{PrivK}_{A,\pi}^{\mathrm{mult}}$
- Note that this multiple-message experiment $\mathrm{PrivK}_{A,\pi}^{\mathrm{mult}}$ is different than $\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}$ defined earlier (indistinguishable encryptions)!
- The end result is that $\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}$ is not very useful as a standalone criterion
  - However, $\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}$ is useful as a building block with formal properties!
- In practice $\mathrm{PrivK}_{A,\pi}^{\mathrm{cpa}}$ is the weakest experiment / definition of interest

***THEOREM 3.21*** *If* $\pi$ *is a (stateless)*[5] *encryption scheme in which* Enc *is a deterministic function of the key and the message, then* $\pi$ *cannot have indistinguishable multiple encryptions in the presence of an eavesdropper.*

[5] Note the ECB is stateless but the rest of the modes presented, including CBC and CTR (and variations w.r.t. the initial vector IV, etc.) are *stateful*.