Notes on Security Analysis of Symmetric Encryption Schemes

Virgil D. Gligor
ENEE 757

1. Symmetric Encryption Schemes

2. Confidentiality Analysis - Example

    - pseudorandom functions and permutations

3. Examples of Symmetric Schemes *proved* Secure

4. Integrity Analysis

5. Examples of Authenticated Encryption Schemes *proved* Secure

Symmetric Encryption - Context

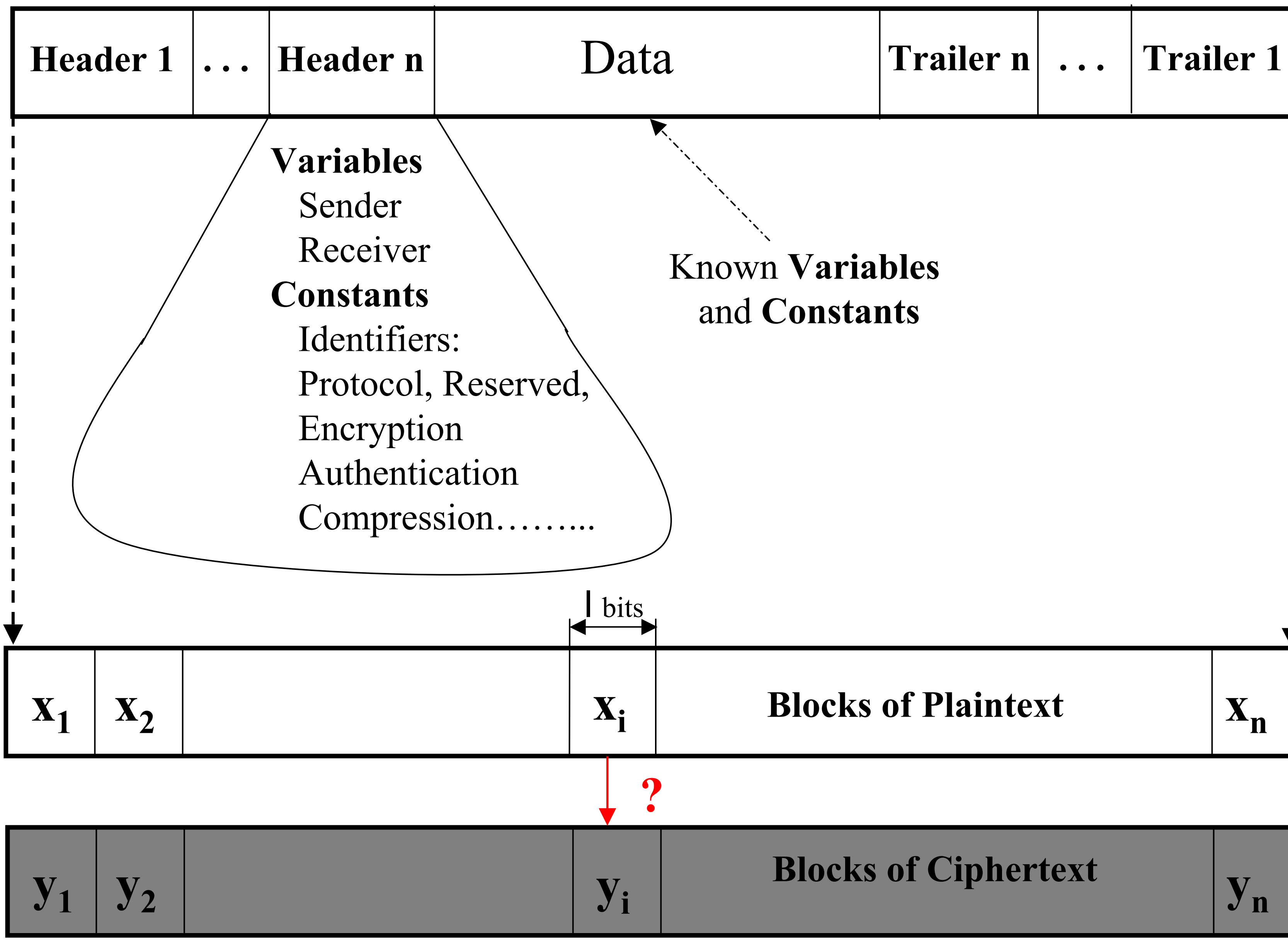
1. Variable Length Messages

2. Fixed-length (Block) Ciphers

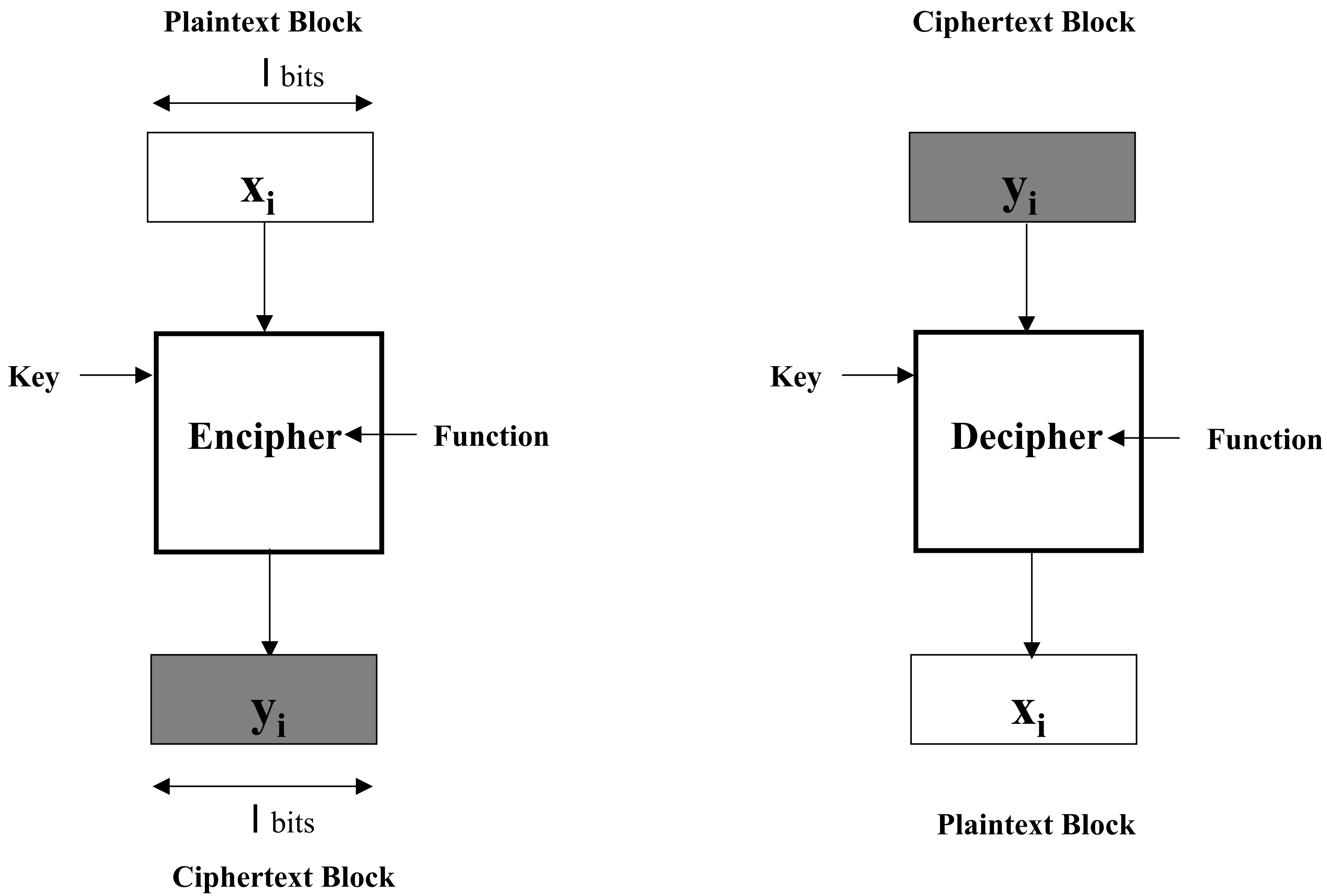
3. Shared Secret Key, K : | K | = k bits

4. Encryption *Schemes* (Modes)

Encryption of Variable-Length Message (after padding)

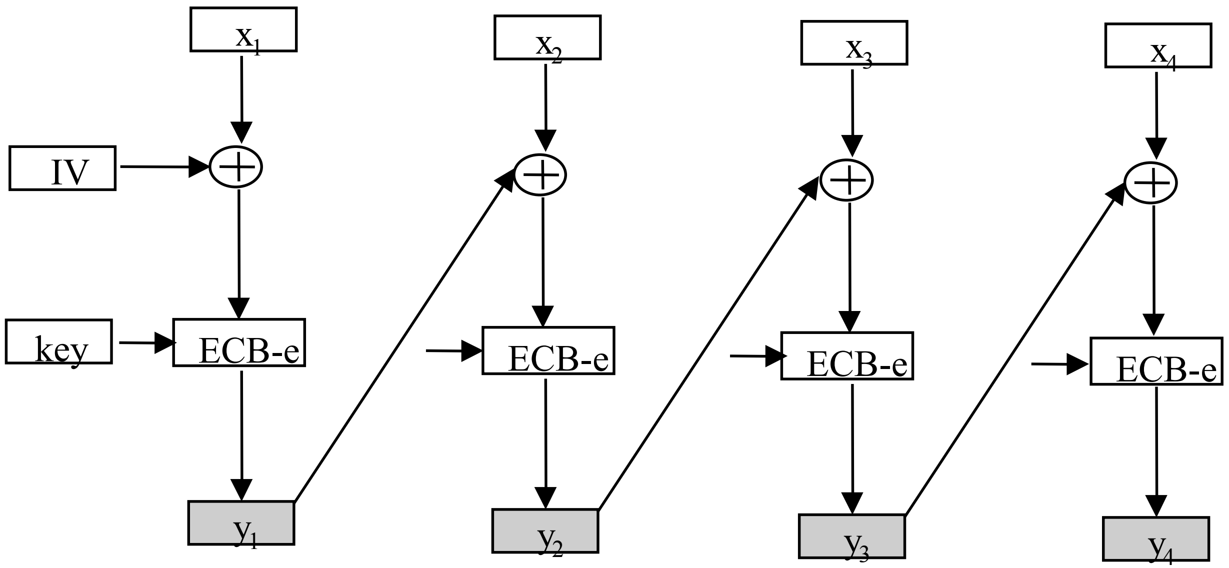


(Fixed-Length) Block Ciphers

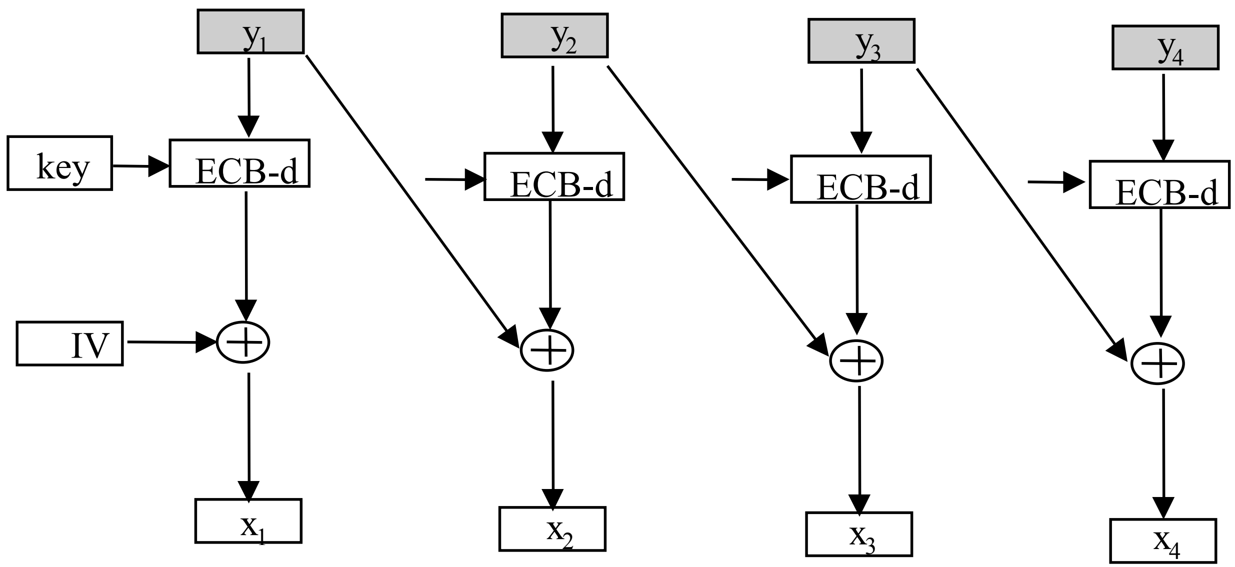


Example of Encryption Mode: Cipher-Block Chaining (CBC)

Encryption : $Y_n = E_K\{Y_{n-1} \oplus X_n\}$, where $Y_0 = IV$

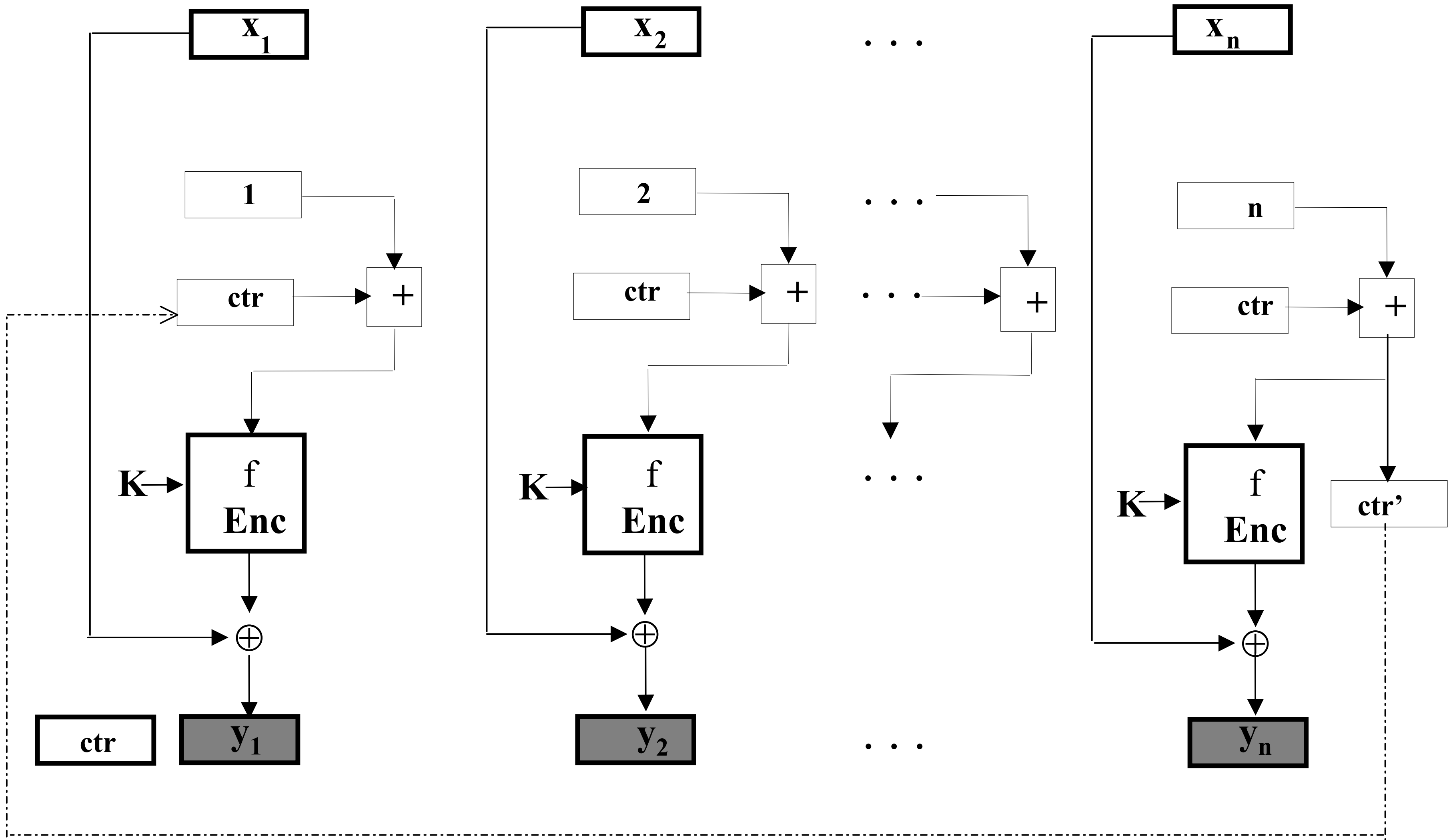


Decryption : $Y_{n-1} \oplus D_K\{Y_n\} = X_n$, where $Y_0 = IV$



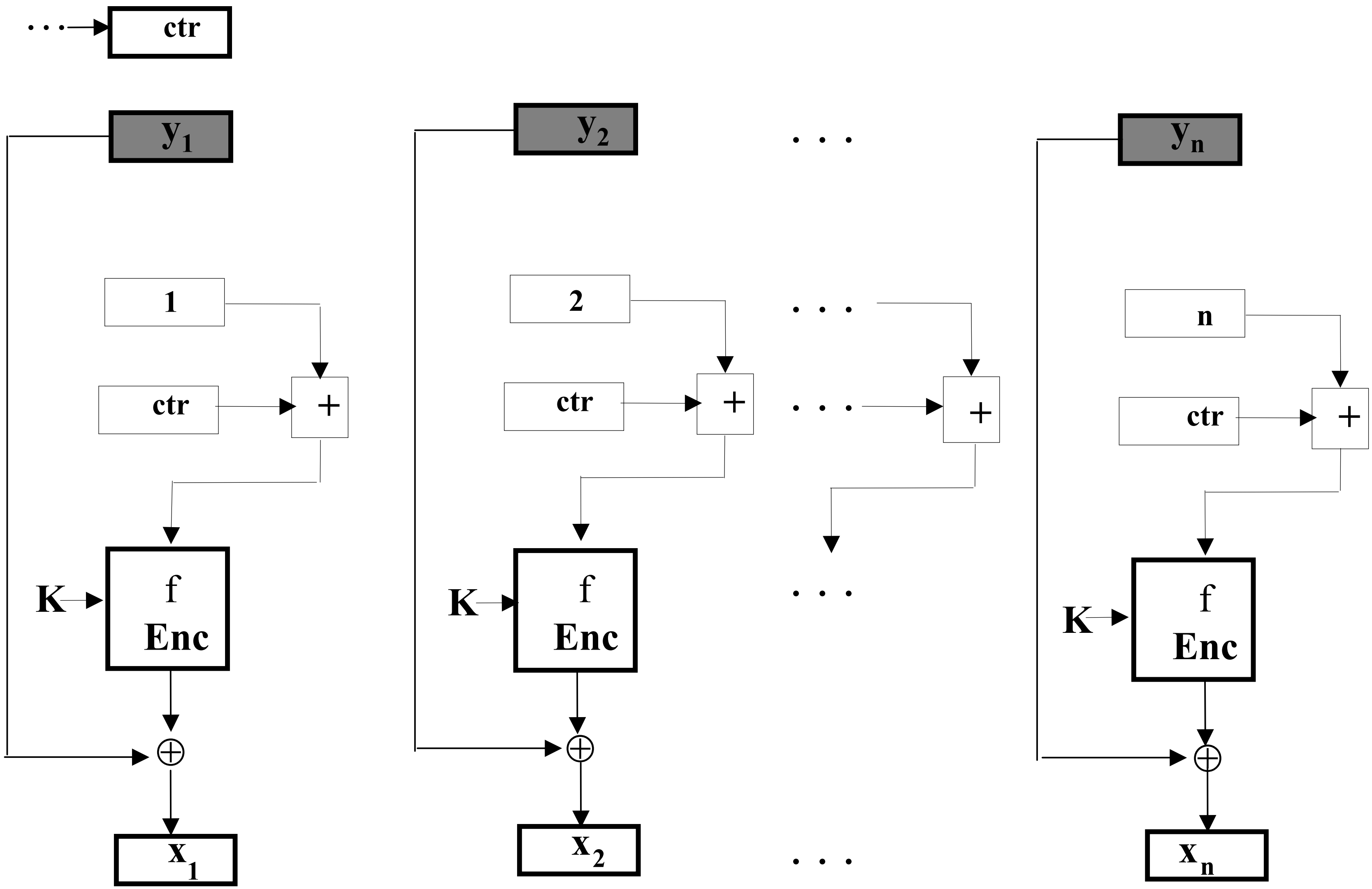
EXAMPLE: Counter-Mode Scheme
XORC - Encryption (BDJR97)*

Initialisation: ctr = -1

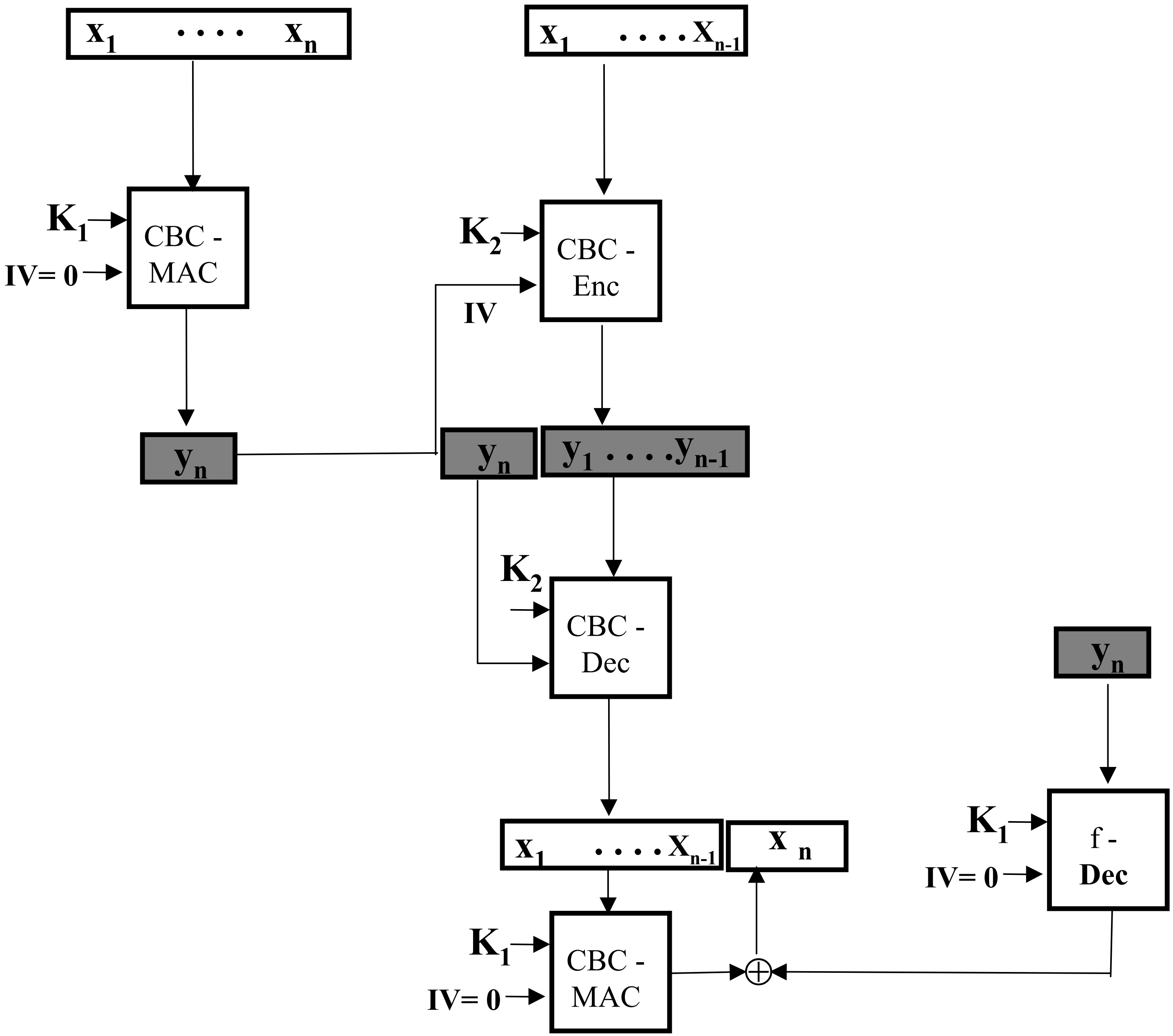


(*) parallel encryption is possible

EXAMPLE: Counter-Mode Scheme ctnd.
XORC - Decryption



EXAMPLE: Two-Pass CBC Scheme (a.k.a VIL cipher)



SECURITY ANALYSIS

3. Is an Encryption Scheme ``Secure'' ?

What is ``security'' (i.e., what *attacks* ?)

- chosen plaintext attacks

- *chosen ciphertext attacks*

How good is ``security'' (I.e., what are the *goals*)?

- indistinguishability

1. Can it be used in practice ?

- …..

2. At what performance cost ?

- …..

3. Is an Encryption Scheme ``Secure'' ?

• **Security of Block Ciphers**
  - **standard set of attacks (e.g., AES certification)**
  - **security parameters (i.,e., workfactors; q,t, μ, ε , key length ?)**

• **Reduction of a Scheme's ``Security'' to that of its Block Cipher**
  - **chosen-plaintext secure schemes**
  - **reduction theorems**

Vulnerabilities of schemes proved secure

• **proofs of security in a model may not hold in other models**

Theory Background

1. Finite Families of Pseudorandom Functions

    - Bellare, Killian, Rogaway (Crypto `94)
    - with roots in earlier work by Golderich, Goldwasser and Micali (JACM 1986)

2. Secure Encryption Schemes - against chosen-plaintext attacks only

    - Bellare, Desai, Jokipii, and Rogaway (STOC 97)

    e.g., real-or-random, left-or-right secure schemes

3. Secure MAC Schemes - against chosen-message attacks

    - Bellare, Guerin, and Rogaway (Crypto `95)
    - Bellare, Canetti, and Krawczyk (Crypto `96), HMAC - IP standard

Finite Families of Pseudorandom Functions and Permutations
(BKR ‘94, BDJR’97)

R : $\{0,1\}^l$ --> $\{0,1\}^L$ - *all* functions that map l-bit strings to L-bit strings

$f_K \in R$; f is identified by key **K** (K is the identifier of the truth table for f)

Use: share **secret key K**, and encrypt / decrypt with $\mathbf{f_K}$ (may use random *permutations* **P**)

Problem: R has a very large number of functions (2^{L2^l}),
and needs very long keys K to identify f_K

=> *family of random functions is impractical*

Solution: Choose a smaller family **F** and make it *look like* **R** (or **P**) *to outsiders*

Finite Families of Pseudorandom Functions and Permutations (ctnd)

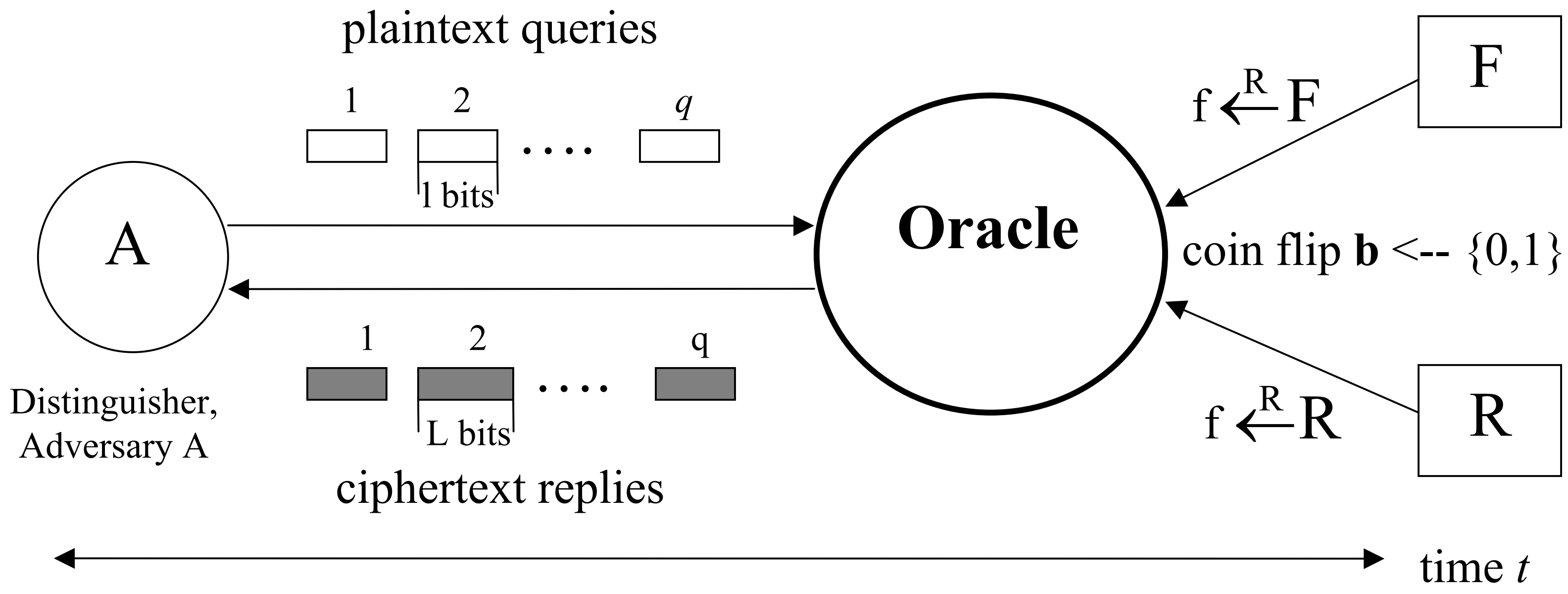
$F_K^k : \{0,1\}^l \dashrightarrow \{0,1\}^L$ - a *set of functions f* that map l-bit strings to L-bit strings
and an associated *set of keys K* <-- $\{0,1\}^k$ of length *k*

function ***f*** is picked *at random* from F_K^k (denoted by $f \xleftarrow{R} \mathbf{F}$) <=>
draw *K* uniformly at *random* from $\{0,1\}^k$ and let $f = F_K$

Let F denote F_K^k

- finite family **F** is pseudorandom if *it looks random* to *outsiders* (i.e., someone who does not know key K)

Finite Families of Pseudorandom Functions (ctnd.)



A's challenge: predict b ($A^f = b$) in q queries and replies and time t (q, t are *large*)

$\Pr[A^f = b] = 1/2 + 1/2 \mathrm{Adv}_A(F,R)$

$$\text{where } \mathrm{Adv}_A(F,R) \triangleq \Pr_{f \xleftarrow{R} F}[A^f = 1] - \Pr_{f \xleftarrow{R} R}[A^f = 1]$$

F is a finite family of PRFs <=> $\mathrm{Adv}_A(F,R) \leq \varepsilon$, where ε is negligible (~**1/q**)

F is (q,t,ε) - pseudorandom *or* (q,t,ε) - secure

F is broken <=> $\mathrm{Adv}_A(F,R) > \varepsilon$

$$\mathbf{Pr\ [A^f = b] = 1/2 + 1/2Adv_A(F,R)}$$

Proof:

$$\Pr[A^f = b] = \Pr[A^f = b \mid b = 1]\Pr[b=1] + \Pr[A^f = b \mid b = 0]\Pr[b=0]$$

$$= \Pr[A^f = b \mid b = 1] \times 1/2 + \Pr[A^f = b \mid b = 0] \times 1/2$$

$$\triangleq \Pr[A^f = 1 \mid b = 1] \times 1/2 + \Pr[A^f = 0 \mid b = 0] \times 1/2$$

$$= \Pr[A^f = 1 \mid b = 1] \times 1/2 + (1 - \Pr[A^f = 1 \mid b = 0)] \times 1/2$$

$$= 1/2 + 1/2(\Pr[A^f = 1 \mid b = 1] - \Pr[A^f = 1 \mid b = 0)])$$

$$= 1/2 + 1/2(\Pr[A^f = 1 \mid f \xleftarrow{R} F] - \Pr[A^f = 1 \mid f \xleftarrow{R} R)])$$

$$\triangleq 1/2 + 1/2(\Pr_{f \xleftarrow{R} F}[A^f = 1] - \Pr_{f \xleftarrow{R} R}[A^f = 1])$$

$$\triangleq 1/2 + 1/2\ Adv_A(F,R)$$

Question:

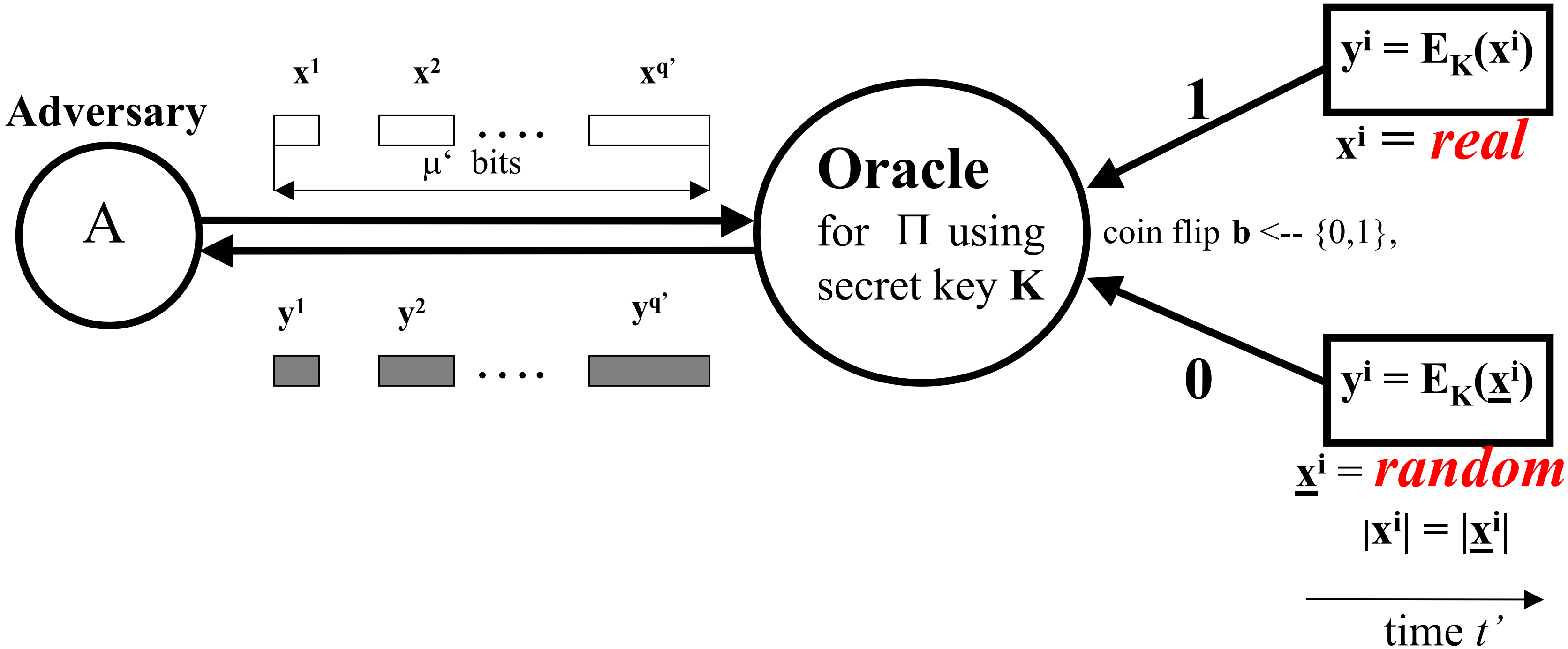
What properties should a mode have to maintain message *secrecy*?

Answer:

It should have an "indistinguishability" property, e.g., in "real-or-random" sense or in a "left-or-right" sense, in an adaptive chosen-plaintext attack (IND-CPA).

=> it must be "probabilistic"

INDinstinguishability-CPA: Secrecy of Scheme $\Pi = (E, D, KG)$



$$Adv^{rr}_{A} = Pr\left[K \leftarrow KG,\ A^{E_K(\,)} = 1\right] - Pr\left[K \leftarrow KG,\ A^{E_K(\_)} = 1\right] \le \varepsilon' \quad \Leftrightarrow$$

$\Pi =$ *(E,D,KG)* is $(q', t', \mu', \varepsilon')$-secure in a ***real-or-random*** (rr) sense

where $(q', t', \mu', \varepsilon')$ are defined in terms of (q, t, ε) of “block cipher” F

Note: equivalent notion of security in a ***left-or-right*** sense is possible

Why Secrecy in the IND-CPA sense ?

IND-CPA (e.g., Real-or-Random) secrecy

=> infeasiblity of recovering

- **the plaintext bits (viz., next example)**
- **XOR of the plaintext bits,**
- **sum of the plaintext bits,**
- **last bit of plaintext,**
- **secret key K**

of a given "challenge ciphertext" in a chosen-plaintext attack

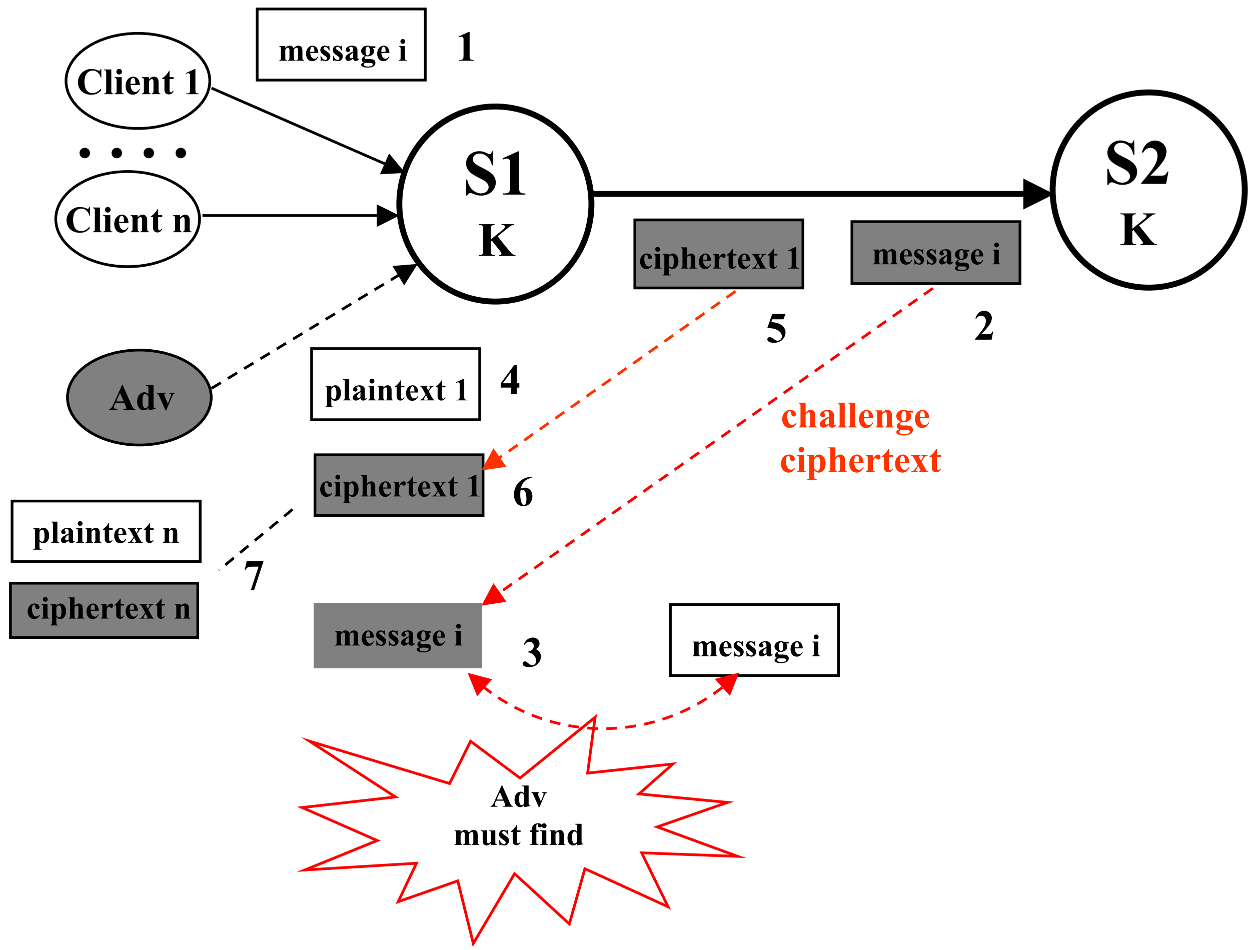
=> Probabilistic Encryption

Answer:

***IND-CPA* security provides a strong notion of secrecy**

Infeasibility of Recovering the Contents of a “challenge ciphetext” in a CPA

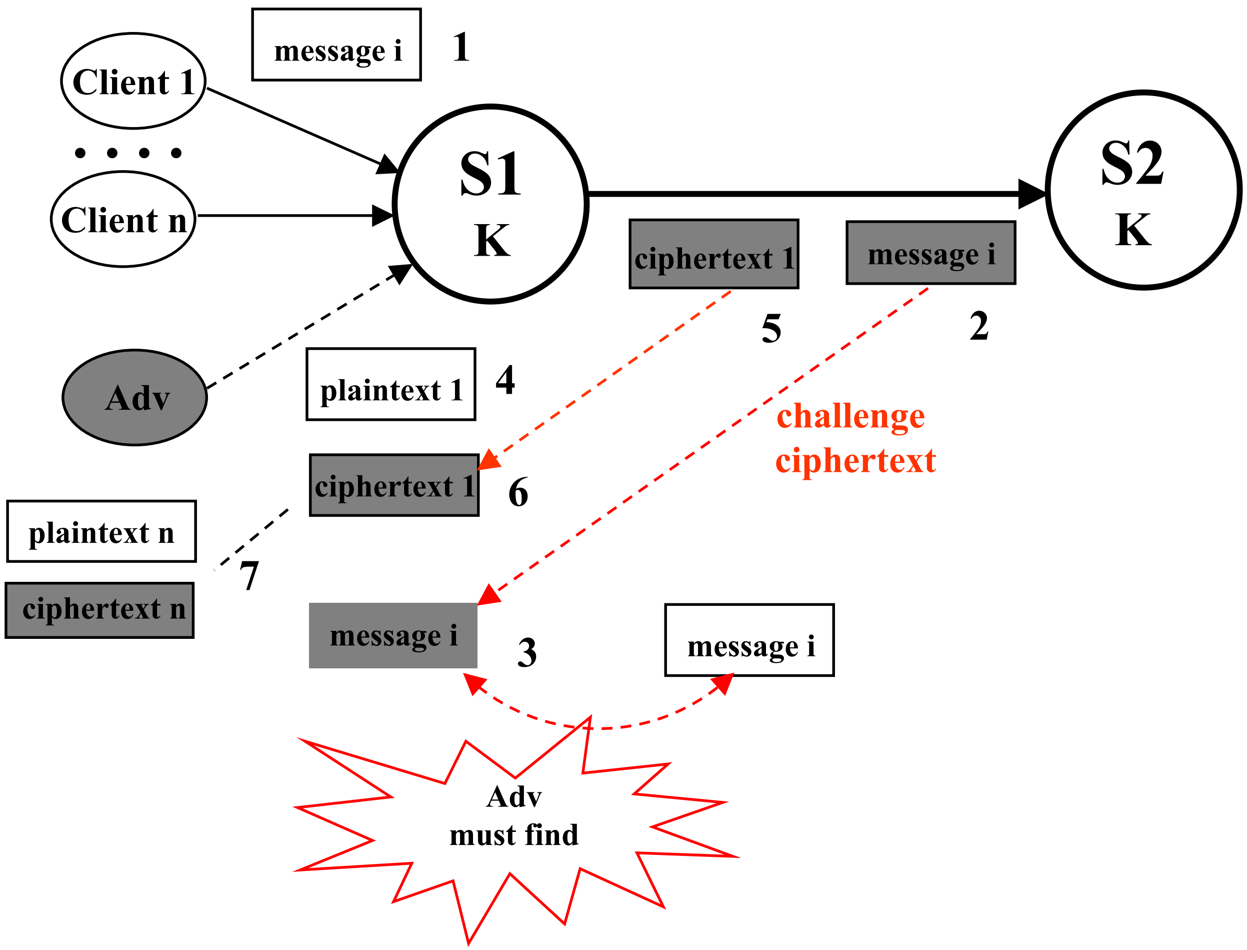
Distributed Service: S (S1, S2), shared secret key K; Clients: Client 1, …, Adv, …,Client n
Adversary: Adv



In attack scenario:
S1 becomes an *Encryption Oracle*

(Intuitive) Secrecy: Infeasibility of Recovering the Contents of a "challenge ciphetext" in a CPA ?

Distributed Service: S (S1, S2), shared secret key K; Clients: Client 1, …, Adv, …,Client n
Adversary: Adv



In attack scenario:
S1 becomes an *Encryption Oracle*

Probabilistic Encryption (Golwasser and Micali 1984)

X = plaintext, $\mathbf{Y_1, \ldots, Y_n}$ = distinct ciphertexts,
$\mathbf{E_K( )} / \mathbf{D_K( )}$ = encryption / decryption with key **K,** and

1. $\mathbf{Y_1 \xleftarrow{R} E_K(X), \ Y_2 \xleftarrow{R} E_K(X), \ldots, Y_n \xleftarrow{R} E_K(X),}$

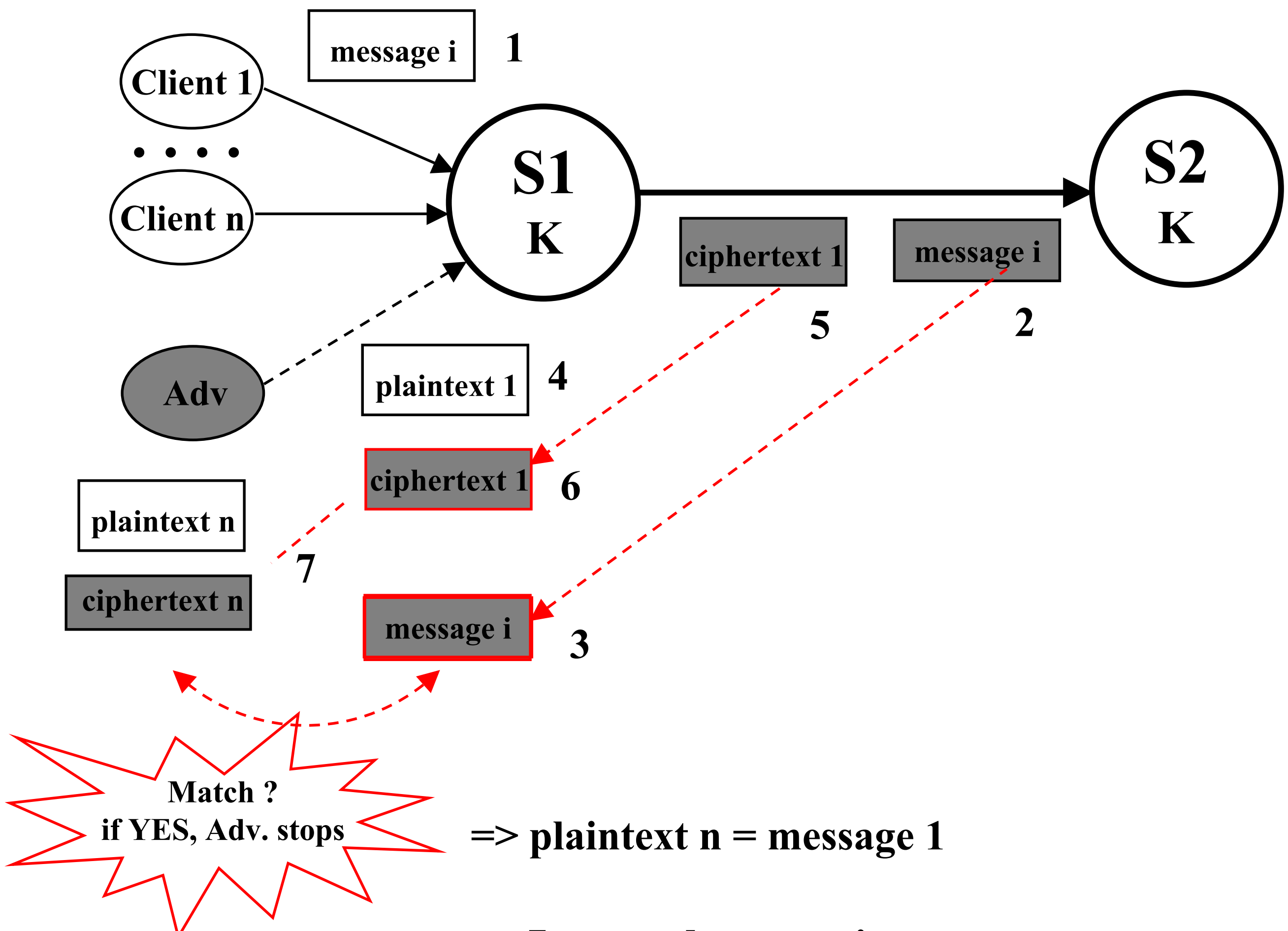
   $\mathbf{X = D_K(Y_1) = D_K(Y_2) =, \ldots, = D_K(Y_n);}$

2. $\mathbf{Y_i \xleftarrow{R} E_K(X)}$ means that
   - $\mathbf{E_k( )}$ picks some **random number**
   - uses the **random number** to compute **Yi**

Why Probabilistic Encryption?

If not, Adv. can Recover the Contents of a (Client's) Challenge Ciphertext in a CPA

Distributed Service: S (S1, S2), shared secret key K; Clients: Client 1, …, Adv, …,Client n
Adversary: Adv



In attack scenario:
S1 becomes an *Encryption Oracle*

We showed that:

Infeasiblity of recovering the plaintext of a given "challenge ciphertext" in a chosen-plaintext attack => Probabilistic Encryption (with chosen plaintexts)

What about :

Real-or-Random **Security => Infeasiblity of recovering the plaintext of a given "challenge ciphertext" in a chosen-plaintext attack ?**

Proof (by contradiction)

Let **B** = an adversary that returns plaintext **X** of challenge ciphertext $\mathbf{Y_{m+1}}$ after choosing plaintexts $\mathbf{(X_1,\ldots,X_m)}$ and receiving corresponding ciphertexts $\mathbf{(Y_1,\ldots,Y_m)}$; i.e., $\mathbf{P_B}$**(success) is non-negligible**

Let $\mathbf{A^O}$ be an **adversary** that is given a **R-or-R oracle O.**
Adversary $\mathbf{A^O}$ performs the following steps;

for i = 1,…, m+1, do
  choose $\mathbf{X_i}$
  obtain $\mathbf{Y_i} \xleftarrow{R} \mathbf{O}(\mathbf{X_i})$
end for
$\mathbf{X} \leftarrow \mathbf{B[(X_1, Y_1), \ldots, (X_m, Y_m), Y_{m+1}]}$
If $\mathbf{X = X_{m+1}}$**, then return 1; else return 0.**

From adversary's $\mathbf{A^O}$ steps, noting that **B** has no information about $\mathbf{X_{m+1}}$, we obtain:

$$\mathbf{Adv^{rr}(A^O) = P_B(success|X_i = real) - P_B(success|X_i = random) \geq P_B(success) - 1/2^n},$$

where n is large

Reduction Proof -- Generic Version

Goal: $\mathbf{Adv_D(F,R)} > \varepsilon$ => $\mathbf{Adv^{ind\text{-}cpa}_A[\Pi\,(F)]} > \varepsilon'$

or how to define of $(q', t', \mu', \varepsilon')$ of Π **in terms of** (q, t, ε) **of F**

Let $\mathbf{Adv^{ind\text{-}cpa}_A[\Pi\,(R)]}$ be the advantage of adversary **A** in breaking a given *scheme* Π in the **real-or-random** (alternatively, in **left-or-right**) sense when the scheme is implemented with **R**

1. *Prove* Π *is secure in an ideal implementation:* $\mathbf{Adv^{ind\text{-}cpa}_A[\Pi(R)]} \leq \delta_R$ (ITLemma)

2. *Contradict Goal:* assume adversary **A** can break the scheme when it is implemented with **F** (which is **known to be a PRF family**); i.e., $\mathbf{Adv^{ind\text{-}cpa}_A[\Pi\,(F)]} > \varepsilon'$

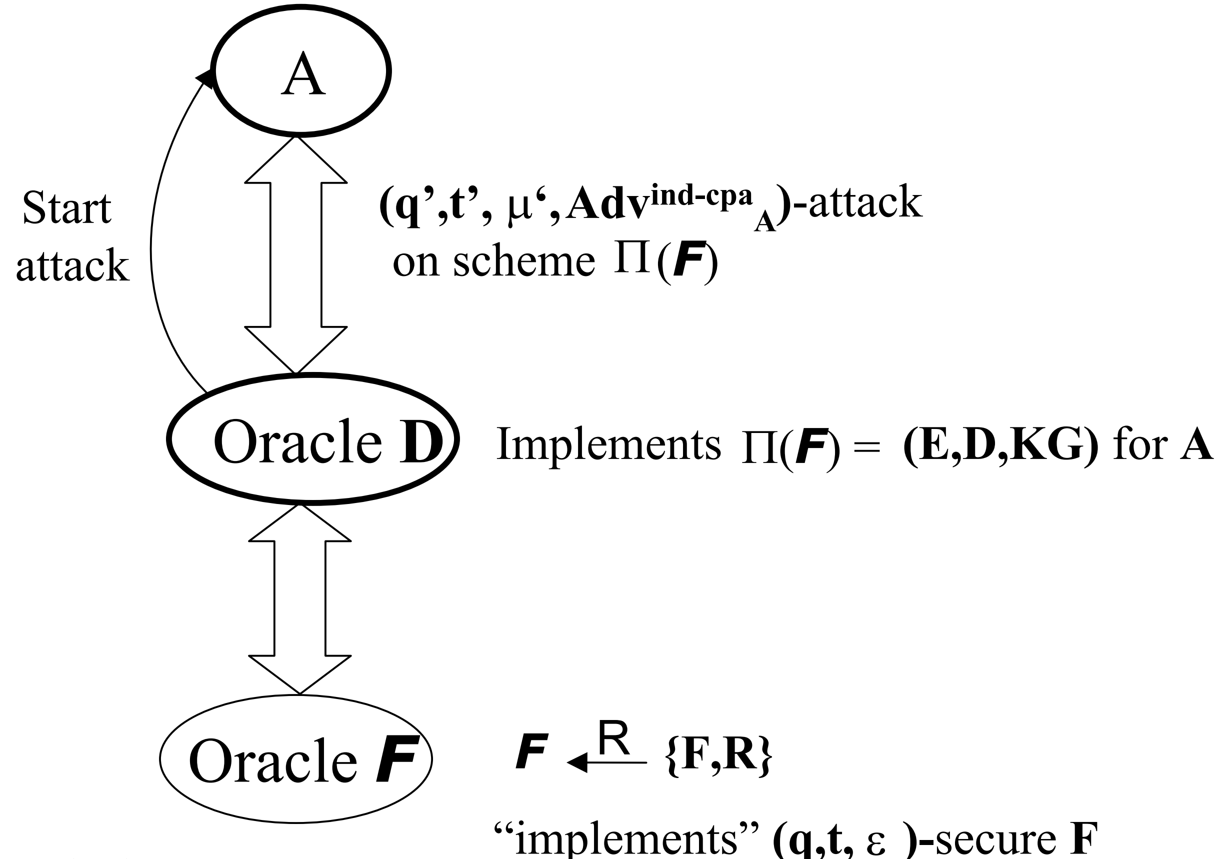
3. *Construct distinguisher* **D** such that

- **D** simulates the scheme Π for **A**'s use
  - using an oracle for the function family $\boldsymbol{F} \xleftarrow{R} \{\mathbf{F,R}\}$
- **D** uses **A** to "break" function family $\boldsymbol{F}$ (under assumption (2))
  (i.e., distinguish **F** vs. **R** with $\mathbf{Adv_D(F,R)} > \varepsilon$)

4. *Prove* that if **D** "breaks" $\boldsymbol{F}$ using adversary **A** that "breaks" Π **(F)** , then a relationship must exist between

$$(q', t', \mu', \varepsilon') \text{ and } (q, t, \varepsilon)$$

Step 3:



1. **D** flips a coin **b** <-- {0,1}
2. Begin
    - **D** runs **A**, and replies to **A**'s queries until **A** stops
        - (1) When **A** makes query **x**:
            - (i) If **b = 1**, **D** encrypts **x** with $\mathbf{E_K}$.
            - (ii) Otherwise, **D** encrypts a random string **x'**, **|x'|=|x|**, with $\mathbf{E_K}$
        - and returns result to **A**.
        - (2) **A** stops making queries, and outputs its guess **c <-- {0,1**}.

   End
3. If **c = b**, **D** outputs **1** (**f** is chosen from **F**); else **D** outputs **0** (**f** is chosen from **R**).

Step 4: Compute $\mathbf{Adv_D(F,R)}$ in D's attack against ***F***

$\mathbf{Adv_D(F,R) = Pr\ [Correct^{ind\text{-}cpa}_{A\ \Pi}(F)] - Pr\ [Correct^{ind\text{-}cpa}_{A\ \Pi}(R)]}$, since **D** "mimics" **A**'s output;

but $\Pr\ [\mathrm{Correct}^{ind\text{-}cpa}_{A\ \Pi}(X)] = 1/2 + 1/2\ \mathrm{Adv}^{ind\text{-}cpa}_{A\Pi}(X)$, where $\mathbf{X} \in \{\mathbf{F,R}\}$ and hence

$\mathbf{Adv_D(F,R) = 1/2\{Adv^{ind\text{-}cpa}_A\ [\Pi\,(F)] - Adv^{ind\text{-}cpa}_A[\Pi\,(R)]\}}$

but $\mathbf{Adv^{ind\text{-}cpa}_A[\Pi\,(R)]} \leq \delta_R$ by Lemma and $\mathbf{Adv^{ind\text{-}cpa}_A[\Pi\,(F)] >} \varepsilon'$ by assumption. Hence,

$\mathbf{Adv_D(F,R)} \geq \mathbf{1/2\{Adv^{ind\text{-}cpa}_A[\Pi\,(F)] -} \delta_R\}$, and

$\mathbf{Adv_D(F,R)\ > 1/2(}\varepsilon' - \delta_R\mathbf{)}$.

If we let $\varepsilon = \mathbf{1/2(}\varepsilon' - \delta_R\mathbf{)}$,

we obtain the desired **contradiction** [i.e., $\mathbf{Adv^{ind\text{-}cpa}_A[\Pi\,(F)] >} \varepsilon' \mathbf{\Rightarrow Adv_D(F,R)\ >} \varepsilon$] , namely that **F** is not **(q, t,** ε **)-PRF** family and relationship $\varepsilon' = 2\varepsilon + \delta_R$

Relationships between q', t' and q,t are obtained by enforcing the related bounds of oracles for ***F*** and **D;** i.e., $\mu' = q'L$, $t' = t - c(l+L)\mu'/L$, where c is a performance constant.

Examples of Encryption Schemes *(E, D, KG)* Proven IND-CPA secure - BDJR97

XORC (stateful, or counter-based XOR a.k.a **CTR mode)**

Initial ctr = 0

function E-XORC^f(x, ctr)
for i = 1,…,n **do** $y_i = f(ctr + i) \oplus x_i$
ctr' <-- ctr + n
return (ctr', ctr$||y_1,\ldots,y_n$)

function D-XOR\$^{f}(z)
Parse z as ctr$||y_1,\ldots,y_n$
for i = 1,…,n **do** $x_i = f(ctr + i) \oplus y_i$
return $x_1,\ldots,x_n$

Note: ctr/ctr' is the current/next state of the counter. For simplicity, assume |x| = nl

Theorem *(Security of XORC using a **PRF**)*
There is a constant c for which the following is true.
Suppose **F** is a (q,t,ε) - secure **PRF** family with input *l* and output *L*. Then for any q the XORC(**F**) scheme is (q',t',μ',ε')- secure in the IND-CPA sense for
$\mu' = q'L$, $t' = t - c(l+L)\mu'/L$, and $\varepsilon' = 2\varepsilon + \delta_R$, where $\delta_R = 0$.

Proof of Theorem

Prove **Lemma** $\mathbf{Adv}^{\text{ind-cpa}}_{\mathbf{A}}\,[\mathbf{XORC(R)}] \le \delta_R \mathbf{= 0}$, and then apply **reduction-proof idea.**

Let adversary **A**: have an L-or-R **oracle** for XORC(**R**)

$(\mathbf{x}_{i,0}, \mathbf{x}_{i,1})$ be the i-th query to the L-or-R oracle

$|x_{i,0}| = |x_{i,1}| = n_i$

Let $\mathbf{y_i}$ = oracle's ciphertext response to **A**'s query i, and **b** be the oracles' coin flip

$$\mathbf{x}_{1,b} = \quad \mathbf{x}_{1,b}[1]\ \mathbf{x}_{1,b}[2] \ldots \mathbf{x}_{1,b}[n_1]$$
$$\mathbf{y}_i = \quad \mathbf{0}\quad \mathbf{y}_1[1]\quad \mathbf{y}_1[2] \ldots \quad \mathbf{y}_1[n_1]$$

$$\in n_1 + \ldots + n_{q-1}$$

$$\mathbf{x}_{2,b} = \quad \mathbf{x}_{2,b}[1]\ \mathbf{x}_{2,b}[2] \ldots \mathbf{x}_{2,b}[n_2]$$
$$\mathbf{y}_2 = \quad n_1\quad \mathbf{y}_2[1]\quad \mathbf{y}_2[2] \ldots \quad \mathbf{y}_2[n_2]$$

. . . .

$$\mathbf{x}_{q,b} = \quad \mathbf{x}_{q,b}[1]\ \mathbf{x}_{q,b}[2] \ldots \mathbf{x}_{q,b}[n_2]$$
$$\mathbf{y}_q = n_1 + \ldots + n_{q-1}\quad \mathbf{y}_q[1]\quad \mathbf{y}_q[2] \ldots \quad \mathbf{y}_q[n_q]$$

$$\text{where } \mathbf{y_i}[j] = f(ctr_i + j) \oplus \left\{ \begin{array}{ll} \mathbf{x}_{i,1}[j], & \text{if } b = 1 \\ \mathbf{x}_{i,0}[j], & \text{if } b = 0 \end{array} \right\} = f(n_1 + \ldots + n_{q-1} + j) \oplus \left\{ \begin{array}{ll} \mathbf{x}_{i,1}[j], & \text{if } b = 1 \\ \mathbf{x}_{i,0}[j], & \text{if } b = 0 \end{array} \right\}$$

Hence, $\mathbf{Adv}^{\text{ind-cpa}}_{\mathbf{A}}\,[\mathbf{XORC(R)}] = \mathbf{Adv}^{\text{l-or-r}}_{\mathbf{A}}\,[\mathbf{XORC(R)}] = \mathbf{0}$, since

- all inputs to f are distinct

- $f \overset{R}{\leftarrow} \mathbf{R}$

Pseudorandom Permutations - Definition

Let $\mathbf{P^l : \{0,1\}^l \text{->} \{0,1\}^l}$ be the family of *all* permutations of l-bit strings to l-bit strings,

$\mathbf{F : \{0,1\}^l \text{->} \{0,1\}^l}$ be the family of ***functions*** of l-bit strings to l-bit strings,

O an oracle for function $\mathbf{g: \{0,1\}^l \text{->} \{0,1\}^l}$

and **D** a distinguisher for **g**; i.e., $\mathbf{g \overset{R}{\leftarrow} F}$ vs. $\mathbf{g \overset{R}{\leftarrow} P^l}$

Goal: make **F** "look like" $\mathbf{P^l}$

Measure how well the goal is reached, by **D**'s advantage:

$$Adv_D(F,P^l) \triangleq Pr_{\mathbf{g \overset{R}{\leftarrow} F}}[D^g = 1] - Pr_{\mathbf{g \overset{R}{\leftarrow} P^l}}[D^g = 1]$$

$Adv_D(F,P^l) \leq \varepsilon$ <=> F is a PRP family

Note: in some analyses we also need *super PRP* families

A Birthday "Attack"

Let $\mathbf{R}_{l,l} : \{0,1\}^l \text{->} \{0,1\}^l$ be the family of *all* functions of l-bit strings to l-bit strings,

$\mathbf{P} : \{0,1\}^l \text{->} \{0,1\}^l$ be a family of permutations of l-bit strings to l-bit strings,

O an oracle for function $\mathbf{g}: \{0,1\}^l \text{->} \{0,1\}^l$

and **D** a distinguisher for **g**; i.e., $\mathbf{g} \overset{R}{\leftarrow} \mathbf{R}_{l,l}$ vs. $\mathbf{g} \overset{R}{\leftarrow} \mathbf{P}$

Goal: find whether $\mathbf{g} \overset{R}{\leftarrow} \mathbf{P}$ or $\mathbf{g} \overset{R}{\leftarrow} \mathbf{R}_{l,l}$ in $\mathbf{2} \leq \mathbf{q} \leq \mathbf{2}^{(l+1)/2}$ queries.

Measure how well the goal is reached, by **D**'s advantage:

$$\mathrm{Adv}_D(\mathbf{P}, \mathbf{R}_{l,l}) = \Pr_{\mathbf{g} \overset{R}{\leftarrow} \mathbf{P}}[D^g = 1] - \Pr_{\mathbf{g} \overset{R}{\leftarrow} \mathbf{R}_{l,l}}[D^g = 1] \geq 0.3\, \frac{\mathbf{q(q-1)}}{\mathbf{2^l}}$$

$$= \mathbf{1 - [1 - C(N,q)]} \geq 0.3\, \frac{\mathbf{q(q-1)}}{\mathbf{2^l}}$$

Background: the "Birthday" Problem (again)

Experiment : throw **q** balls, at random, into **N** buckets; $N \geq q$

Problem: Find bounds on

C(q,N) = probability of "collisions" of balls in buckets

(i.e., probability of at least two balls in same bucket)

Facts:

(1) $C(q,N) \leq \frac{q(q-1)}{2N}$

(2) $C(q,N) \geq 1 - e^{\frac{q(q-1)}{2N}}$

(3) for $1 \leq q \leq (2N)^{1/2}$

$$C(q,N) \geq 0.3 \frac{q(q-1)}{N}$$

Example: q = 23 people, N=365 days/year => C(23, 365) > 1/2

probability that at least 2 persons in a room of 23 people have same birthdate > 1/2

100 > 0.99

Using PRP families (instead of PRF families) as Block Ciphers

Motivation:

(1) Few encryption modes can use PRF families since most modes need to use f^{-1} for decryption

[but one can encrypt more with PRF families since birthday attacks are not possible; e.g., XORC (CTR-mode)]

(2) However, it is simpler to analyze encryption modes using PRF families

But,

can we do the analysis using PRF families and then modify the bounds as if PRPs were used ?

Using PRP families (instead of PRF families) as Block Ciphers (continued)

Let

$\mathrm{Adv}_D(\mathbf{P}, \mathbf{R}_{l,l}) \triangleq$ (in)security of **P** vs. $\mathbf{R}_{l,l}$

and

$\mathrm{Adv}_D(\mathbf{P}, \mathbf{P}^l) \triangleq$ (in)security of **P** (or **F**) vs. $\mathbf{P}^l$

Then, it can be shown that

$$\mathrm{Adv}_D(\mathbf{P}, \mathbf{R}_{l,l}) \leq \mathrm{Adv}_D(\mathbf{P}, \mathbf{P}^l) + \frac{q(q-1)}{2^{l+1}}$$

That is, the insecurity of a family of permutations P in the PRF sense is greater than that of P in the PRP sense but only by $\frac{q(q-1)}{2^{l+1}}$.

Another Encryption Schemes *(E, D, KG)* Proven IND-CPA secure (ctnd)

CBC ($=stateless)

function E-CBC$^{f}(x)
y_0 <-- (0,1}^{l}
for i = 1,…,n **do** $y_i = f(y_{i-1} \oplus x_i)$
return $y_0||y_1,\ldots,y_n$

function D-CBC$^{f}(z)
Parse z as $y_0||y_1,\ldots,y_n$
for i = 1,…,n **do** $x_i = f^{-1}(y_{i-1}) \oplus y_{i-1}$
return $x_1,\ldots,x_n$

Theorem *(Security of CBC$ using a **PRF**)*
There is a constant c for which the following is true.
Suppose F is a (q,t, ε) - secure PRF family with in put *l* and output *L.* The for any q the CBC$(F) scheme is (q',t', μ', ε')- secure in a left-or-right sense for
$\mu' = q'l$, $t' = t - c\,\mu'$, and $\varepsilon' = 2\varepsilon + \delta_R$ where $\delta_R = (\mu'^2/l^2 - \mu'/l)2^{-l}$

Note 1: We need to adjust the result for this for use of PFPs in practice
(or else we cannot decrypt)
Note 2: This scheme is not (intended to be) secure against forgeries in chosen-plaintext attacks.
Example: Message Splicing and Decomposition invariant of CBC

Examples of Asymptotic Vulnerabilities

(1) Highly formatted messages: constant value at the same, known position

- headers containing protocol and other identifiers
  - WWII messages used by German navy
    - sender and receiver identifiers; e.g., name, rank, unit; *Offizier*
  - Kerberos tickets
  - TCP headers inside IP datagrams

Consequence: exhaustive key table attack against XORC keys

Does the key size, **k**, matter ?

(2) Highly predictable plaintext generated by forged ciphertext

Consequence: need collision-free function to add redundancy for protection against message forgeries

Performance Problem => questionable use

No theory for integrity of encrypted messages !

Consequence: exhaustive key table attack against XORC keys

=> x_i is *known* in a large number of messages (e.g., 2^p) encrypted in different keys

- < ctr+ i, $f_{Ki}(ctr+i)$>, i = 1,…, 2^p, are known in the XORC scheme
- adversary computes table entries $f_{K1}(ctr+i)$, $f_{K2}(ctr+i)$, … , $f_{Km}(ctr+i)$; $m=2^k$
- adversary searches for the 2^p values of $f_{Ki}(ctr+i)$ in table
- a match, and its corresponding key, is found in less than 2^{k-p-1}probes on avg.

=> x_i is *predictable* in a large number of messages (e.g., 2^p) encrypted in different keys

- x_i : $\{x^1_i, x^2_i \ldots, x^r_i\}$ for some small value of r
- adversary searches the table for < $f_{Ki}(ctr+i) \oplus x_i \oplus x^j_i$ > for j = 1,…,r values / key

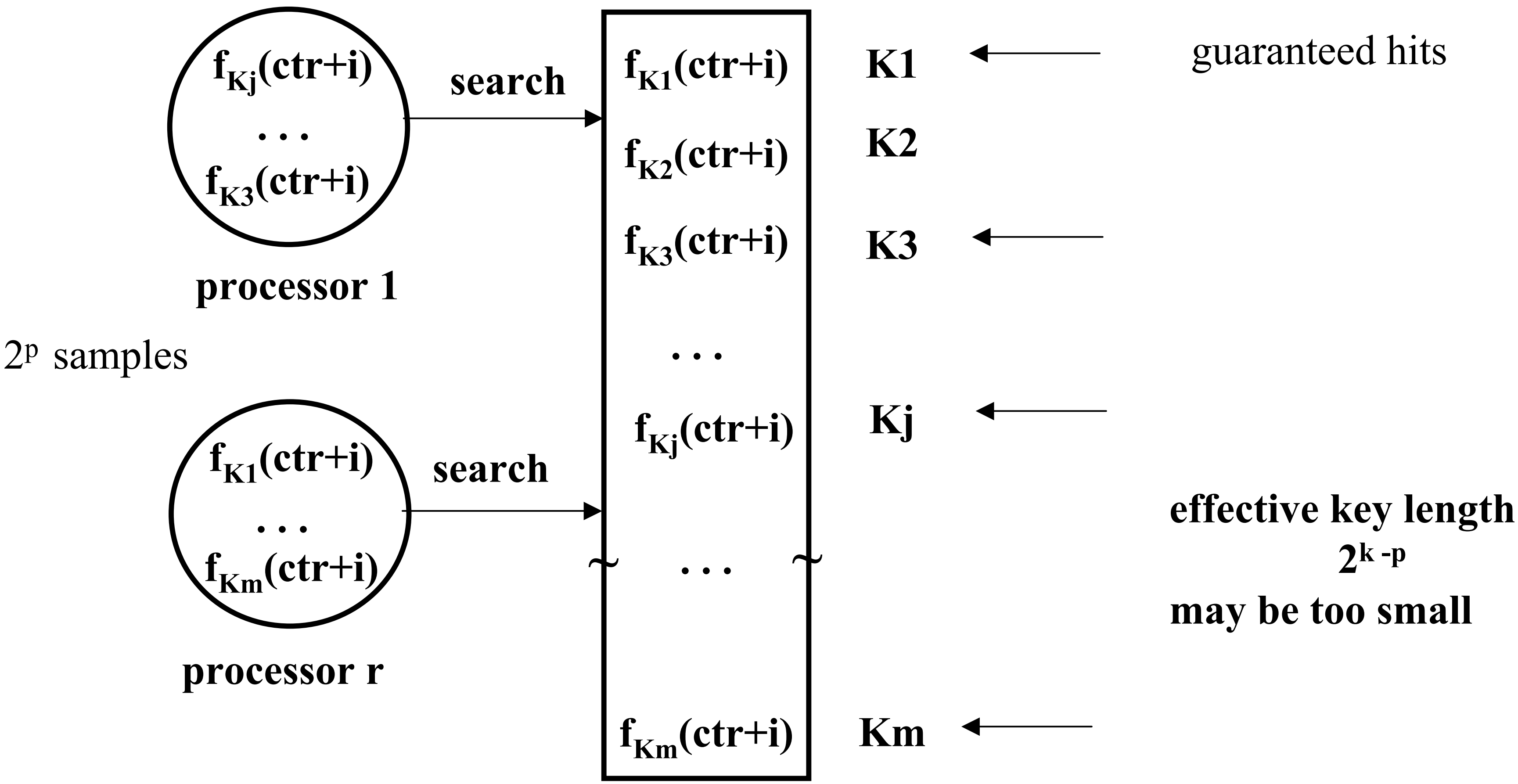
=> back traffic attacks

Consequence: use collision-free function to add redundancy for protection against message forgeries

=> ciphertext bit modification in position i causes plaintext bit modification in position i

Vulnerability 1: Parallel, Exhaustive Key Table Attack (XORC)

x_i is **known** => $<x_i, f_K(ctr+i) \oplus x_i>$ is known, and
ctr is public = $<ctr+i, f_K(ctr+i)>$ is known
xi is **constant** => single-table search

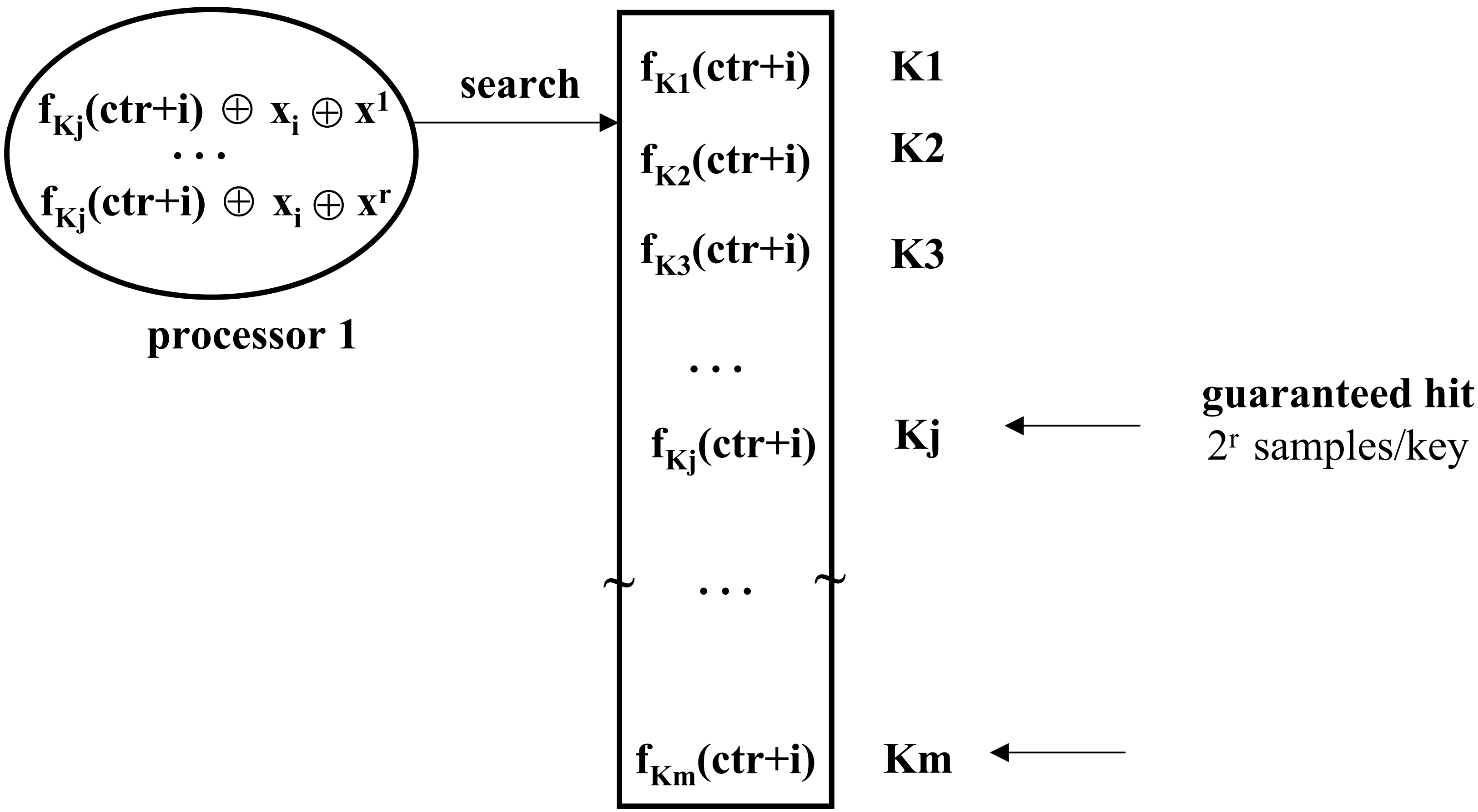


$m = 2^k$ entries, $k = |K|$,
need not be built all at once or in real time

Key length matters, again !

Vulnerability 1: Parallel, Exhaustive Key Table Attack (XORC ctnd)

x_i is *predictable* => $<x_i, f_K(ctr+i) \oplus x_i \oplus x^j>$ **j=1,…,r predicted** values
r searches per key



amount of extra work is a *linear* function of the quality of the prediction

A Solution to *Asymptotic* Vulnerability: Symmetric Encryption with *Random* Counters

Random Counters

Initial value: rctr <-- $\{0,1\}^1$, for every new key or key pair

Counter ``tick" and range: rctr +1 ,…, rctr + 2^l

Per-block, or per-message, tick

Counter values are secret; sequence is not random

Example: XORC Scheme with Random Counters

rctr = per-block random counter

function E-XORC$^{f_{K1} f_{K2}}$(x, rctr)

for i = 1,…,n **do** $y_i = f_{K1}(rctr+i) \oplus x_i$

y_0 <-- $f_{K2}(rctr)$

rctr <-- rctr +n

return $y_0 || y_1, \ldots, y_n$

function D-XOR\$$^{f_{K1} f_{K2}}$(z)

Parse z as $y_0 || y_1, \ldots, y_n$

rctr <-- $f^{-1}_{K2}(y_0)$

for i = 1,…,n **do** $x_i = f_{K1}(rctr+i) \oplus y_i$

return $x_1, \ldots, x_n$

Known or predictable plaintext, back traffic recording no longer helps much

Short keys (e.g., 56 - 64 bit) can be as good as long/very long (e.g., 80/128 bit) keys

Message Integrity Concerns

Message Authentication
- Origin; Content

Message Integrity
- Detect all message modifications (e.g., forgeries) with high probability

Traditional Solutions

- use hash functions, MACs
  => performance (two passes) ; additional crypto primitive
- non-cryptographic MDC functions =>
  inadequate security (i.e., message integrity *and* secrecy)

Old Performance Examples (J. Touch 1995 + update)

Hash Functions	Sparc 20/71 (Mbps)	Sparc 20/61 (Mbps)	Hardware Speedup	Ops/32 bits
• MD5	57	38	x 4	40 - 50
• SHA	30			
• UMAC (fastest MAC to date - peak speed 0.5 cycle / byte				
Checksums				
• IP v4		260	x 5	
• xor *op*				1-2
Block Encryption				
• DES	20.6		x 50	~ 190 (?)
IP v4 (on ATM)	120			

Newer Hash functions: 2 - 10 x MD5 performance

- highly optimized assembly: 2 - 3 performance of C/C++ implementations

Hash functions always have much lower performance than MDC functions

(In) Security Examples

No secure Authenticated Encryption Schemes using non-cryptographic MDC existed before January 2000

Integrity (Authenticity)

0. Authenticated encryption: security definitions and motivation

1. CBC-XOR: An old (failed) attempt at authenticated encryption

2. Perspective: other past (failed) attempts

3. A recent (failed) attempt: NSA's Dual Counter Mode

4. Examples of "provably secure" authenticated encryption modes:
   XCBC-XOR, XECB-XOR(Gligor and Donescu)
   IACBC, IAPM (C.S. Jutla, IBM Research)
   OCB (P. Rogaway, U.C. Davis)

5. Status

Question:

How do we encrypt variable-length messages with block ciphers such that

message *secrecy* and *integrity*

are maintained ?

Answer:

(1) we “Encrypt-then-Authenticate,” or

“Authenticate-then-Encrypt,” or

“Authenticate-and-Encrypt”

(2-passes, possibly 2 cryptographic primitives; power ? performance?)

(2) we use *authenticated encryption* modes

(1-pass, 1 cryptographic primitive; e.g., block cipher+ non-crypto MDC)

Question:

What properties should a mode have to maintain message *integrity*?

Answer:

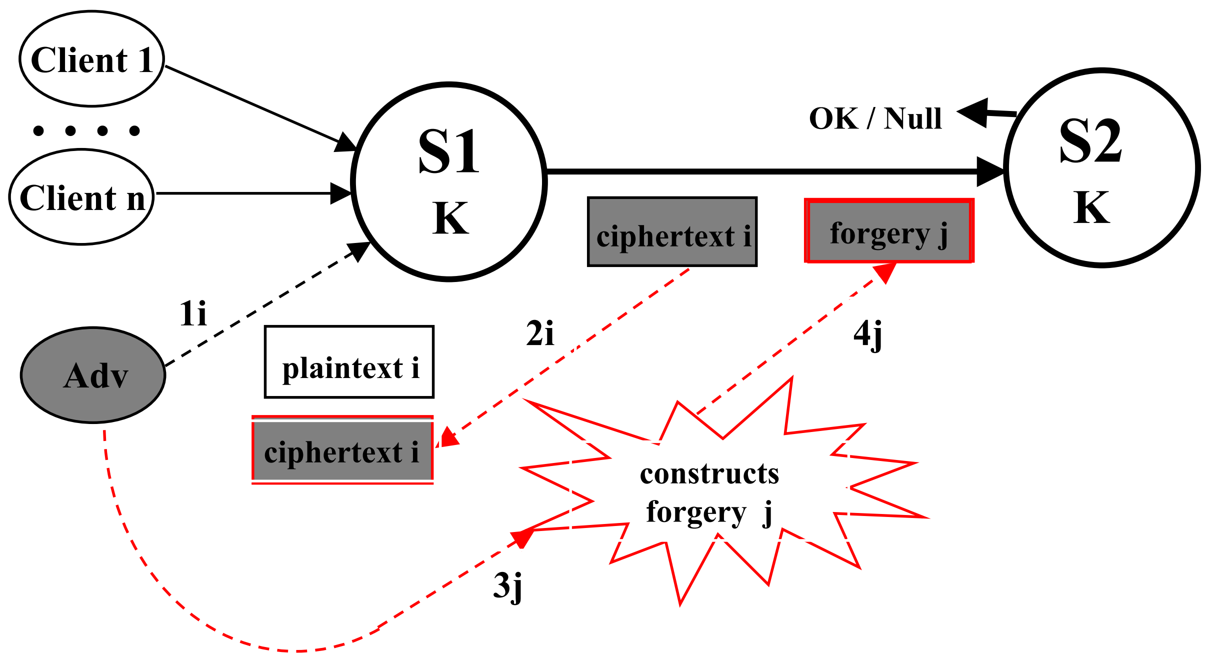
It should protect against "existential forgeies" in chosen plaintext attacks (EF-CPA).

=> it must be "probabilistic"

(but weaker notions exit that might still be useful in practice)

Why Existential-Forgery protection in a CPA? If not, Adv. can construct a valid forgery

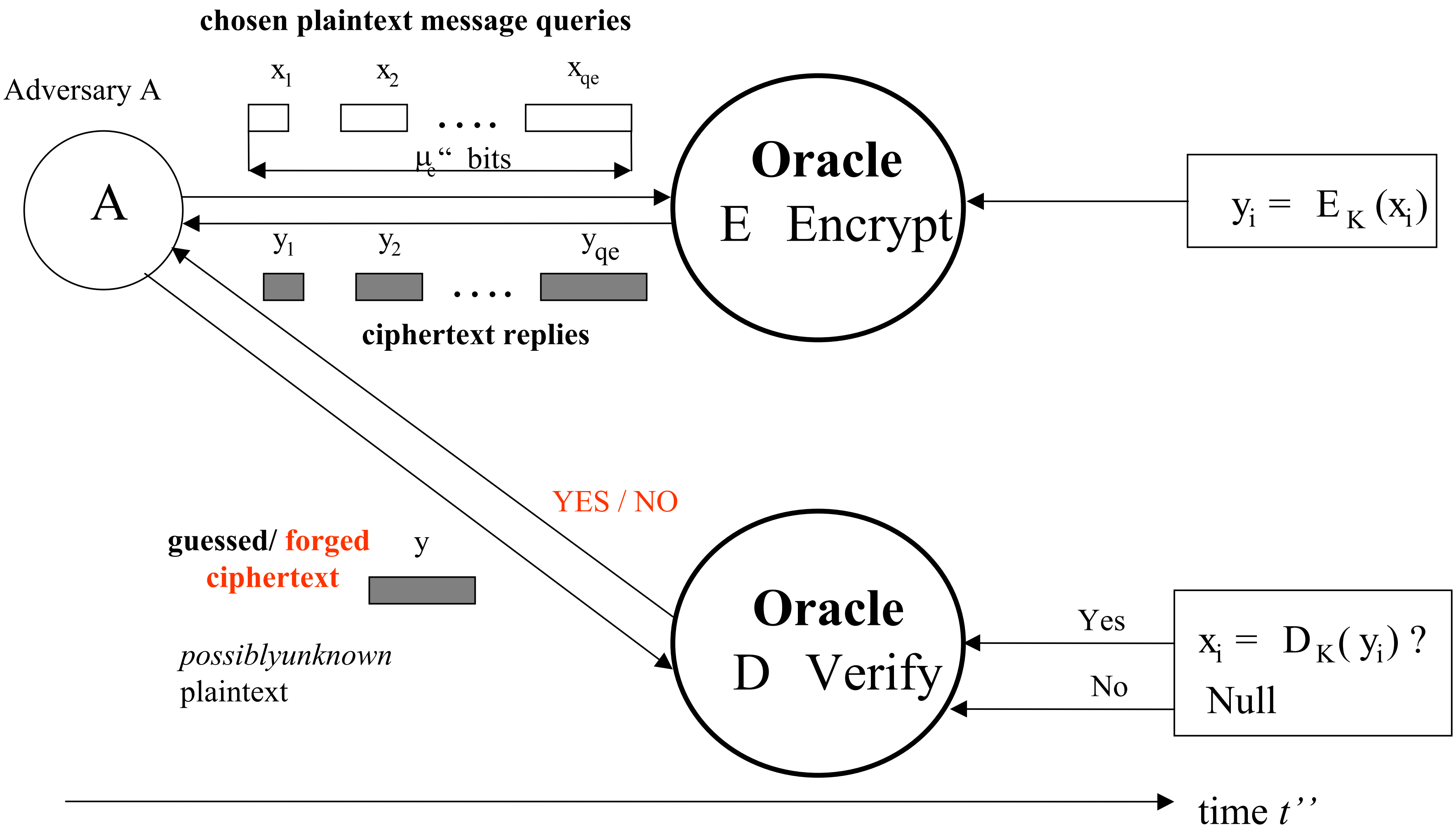
Distributed Service: S (S1, S2), shared secret key K; Clients: Client 1, …, Adv, …,Client n
Adversary: Adv



Why probabilitic ? If not, Adv. Can construct a valid forgery (viz., NSA's Dual Counter Mode)

In attack scenario:
S1 becomes an *Encryption Oracle*
S2 becomes a *Decryption Oracle*

Forgery in Chosen-Plaintext Attack against Scheme ***(E , D , KG)***

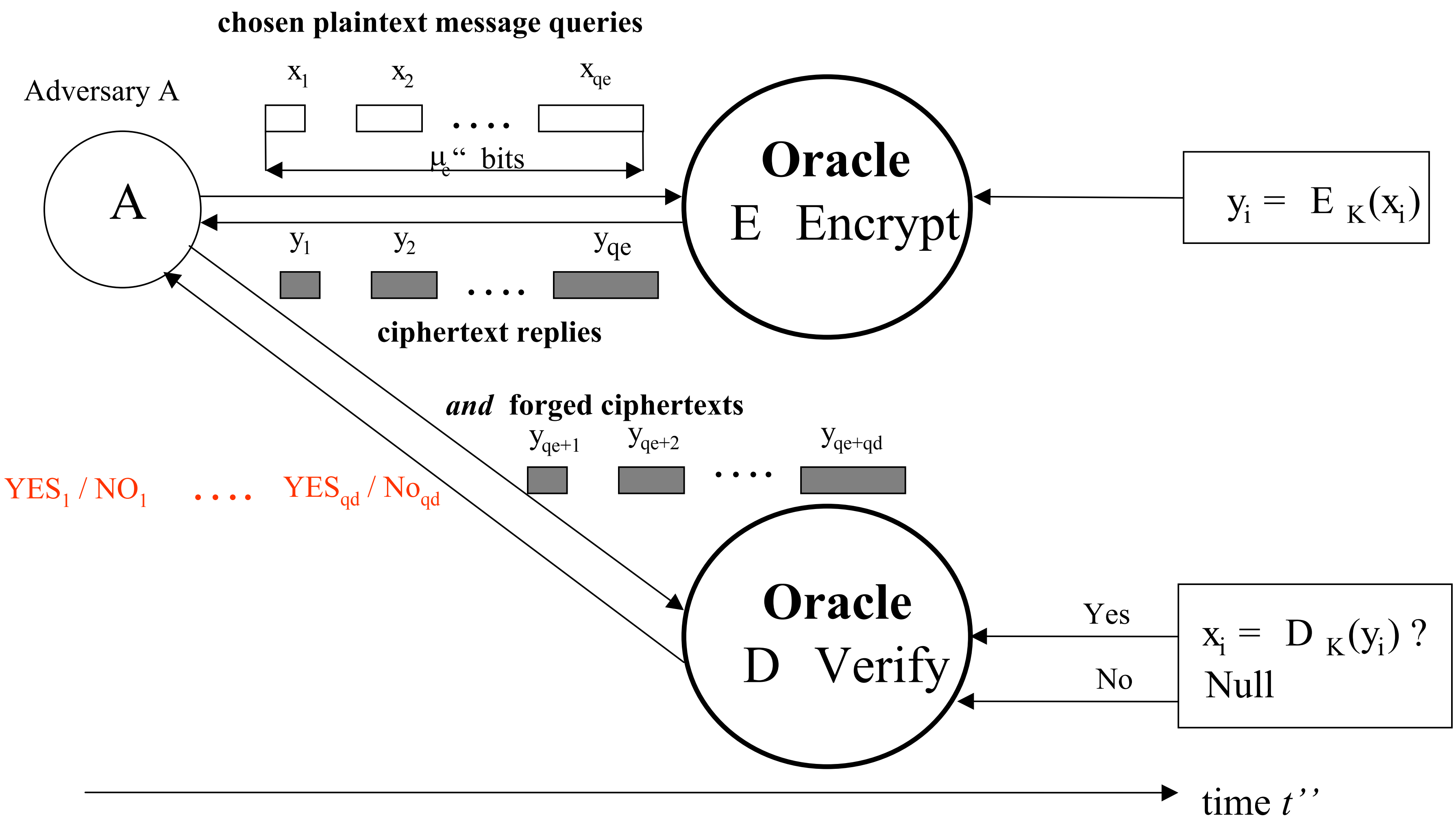


$q'' = qe+qd$

A similar attack can be defined for MAC scheme mMAC

$\mu'' = \mu_e'' + \mu_d''$

Multiple Forgeries in Chosen-Plaintext Attacks



$q'' =$ qe+qd

A similar attack can be defined for MAC scheme mMAC

$\mu'' = \mu_e'' + \mu_d''$

Typical Approach to Authenticated Encryption

1. Partition Message into Blocks

- use padding if necessary

2. Compute Redundancy Block

- use Manipulation Detection Code (*MDC*)

3. Add redundancy block to message blocks

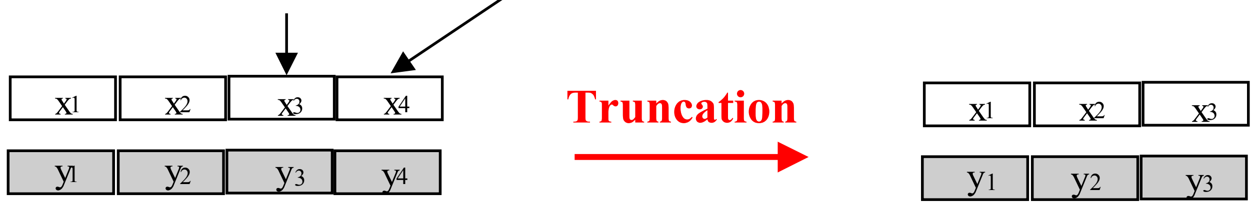
4. Encrypt message and redundancy block

Ex. Integrity (Authentication) Problems of CBC - XOR (and PCBC-XOR)

Forgeries with known plaintext

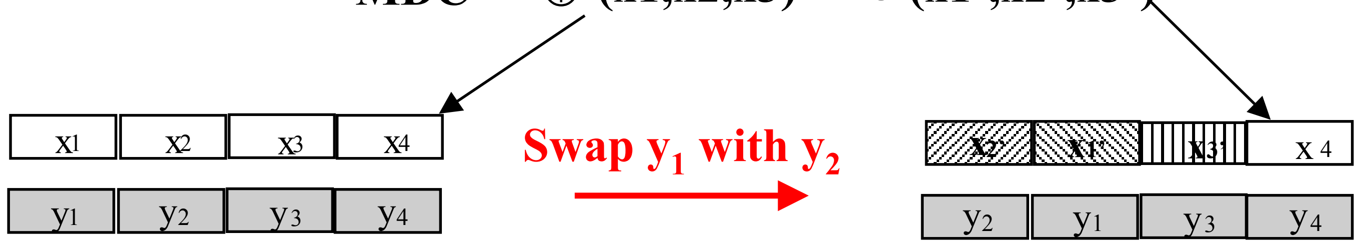
MDC = $\oplus$(x1,x2,x3)

choose x3 = x1 $\oplus$ x2



forgery 1

MDC = $\oplus$ (x1,x2,x3) = $\oplus$ (x1',x2',x3')



forgery 2

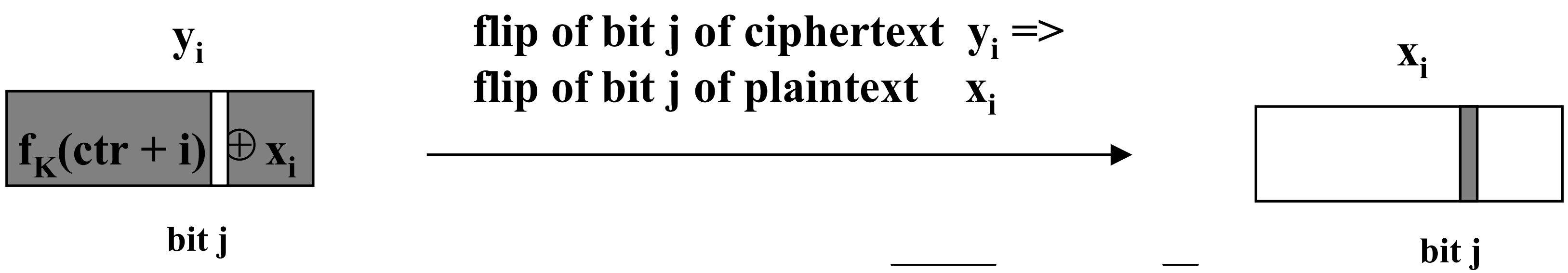
Forgery [with known plaintext *if pair (x,y) is known*]

MDC = $\oplus$(x1,x2,x3) = $\oplus$ (x1, x2, x', x",x3')

forgery 3

Example of Integrity Problems of the XOR Schemes

Forged Ciphertext with Chosen Plaintext outcome



⊕ property used: $\overline{A \oplus B} = A \oplus \overline{B}$

(non-cryptographic MDCs will not detect such attacks)

Past (Failed) Attempts to Provide Authenticated Encryption

1. C. Weissman: use *CBC* with *MDC* = *Cyclic Redundancy Code (CRC)*

- proposed at 1977 DES Conference at NBS
- broken by S. Stubblebine and V. Gligor (IEEE Security and Privacy 1992)

2. C. Campbell: use *Infinite Garble Extension* (IGE) mode with *MDC* = *constant appended to message*

- proposed at 1977 DES Conference at NBS
- IGE was reinvented *at least* three times since 1977
- broken by Gligor and Donescu 1999

3. V. Gligor and B. Lindsay: use *CBC* with *MDC* = *any redundancy code*

- Object Migration and Authentication, IEEE TSE Nov, 1979 (and IBM Research Report 1978)
- known to be broken by 1981 (see below)

4. US Dept. of Commerce, NBS Proposed Standard: Use *CBC* with *MDC* = *XOR*

- withdrawn in 1981; see example of integrity breaks above

Past (Failed) Attempts to Provide Authenticated Encryption (ctnd)

5. MIT Kerberos v.4: use *PCBC* with *MDC = constant appended to last block*

- **proposed at 1987 - 1989**
- **broken by J. Kohl at CRYPTO '89**

6. MIT Kerberos v.5 (1991 ->) use *CBC* with *MDC = confounded CRC-32*

- **confounder (i.e., unpredictable block) prepended to message data**
- **CRC-32 is computed over the counfounded data and inserted into message before encryption**
- **proposed in 1991 Kerberos v.5 specs. (used within US DoD ?)**
- **broken by S. Stubblebine and V. Gligor (IEEE Security and Privacy 1992)**

7. V. Gligor and P. Donescu: use *iaPCBC* with *MDC = unpredictable constant appended as the last block of message*

- **proposed at the 1999 Security Protocols Workshop, Cambridge, UK.**
- **actually the proposal had MDC = XOR**
- **broken first by the "twofish gang" (D, Whiting, D. Wagner, N. Ferguson, J.Kelsey)**

8. US DoD, NSA: Use *Dual Counter Mode* with *MDC = XOR*

- **proposed August 1, 2001 and withdrawn August 9, 2001**
- **broken by P. Donescu, VD. Gligor, D. Wagner and independently by P. Rogaway**

Observations:

1. ***The fastest, surest way to get oneself in the cross-hairs of everyone's loaded rifle is to propose a new mode of encryption.***

2. ***Everyone who has ever proposed an encryption or an authentication mode has gotten at least one wrong, at least once.***

3. Paul van Oorschot, March 1999:

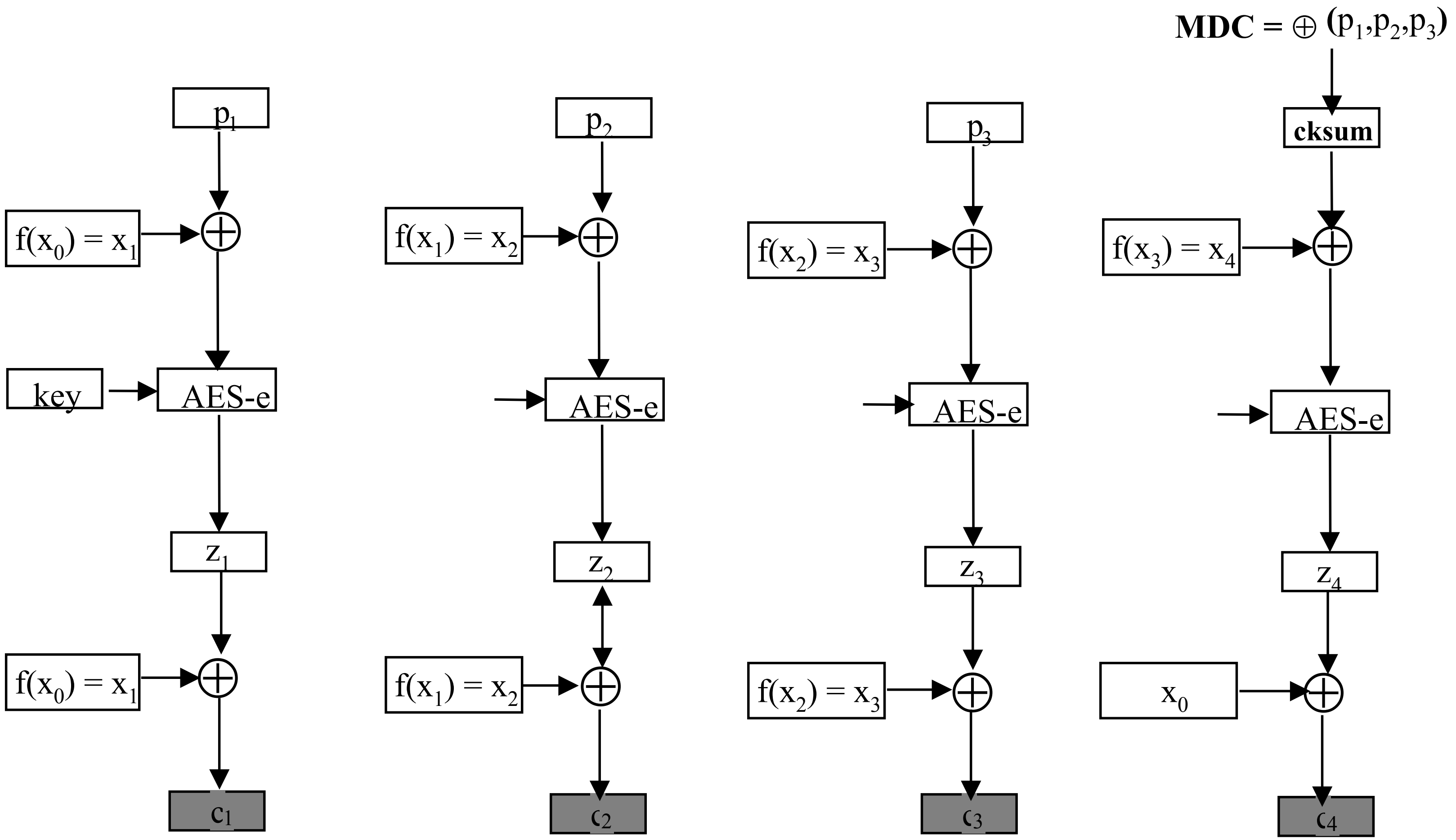
"no one said this was an easy game !"

4. Folklore :

"*Good judgement comes from experience, and experience comes from bad judgement*"

A recent example: NSA's Dual Counter Mode - Version 1

f = connection polynomial of degree W of a LFSR (W = width of block cipher)
$\mathbf{x_0}$ = "shared secret negotiated during key exchange"
$\mathbf{x_0}$ is not (cannot be) generated randomly per message => encryption is not probabilistic
$\mathbf{x_i = f(x_{i-1})}$, **i = 1,…, n+1**; $\mathbf{p_i}$ = plaintext block, $\mathbf{c_i}$ = ciphertext block



Attacks against the Dual Counter Mode - Version 1

Integrity

1. **Since x_0 is not generated per-message (and encryption is not probabilistic),**
   choose $P = p_1 p_2, \ldots, p_n$ such that $p_1 \oplus p_2 \oplus, \ldots, \oplus p_{n-1} = 0$
   and $Q = q_1 q_2, \ldots, q_{n-1}$ such that $q_i = 0;\ i = 1, \ldots n-1.$

   Obtain ciphertexts $C = c_1 c_2, \ldots, c_{n-1}\, c_n\, c_{n+1}$ for **P** and
   $D = d_1 d_2, \ldots, d_{n-1}\, d_n$ for **Q**; then

   $C' = c_1 c_2, \ldots, c_{n-1}\, d_n$ is a *valid forgery*

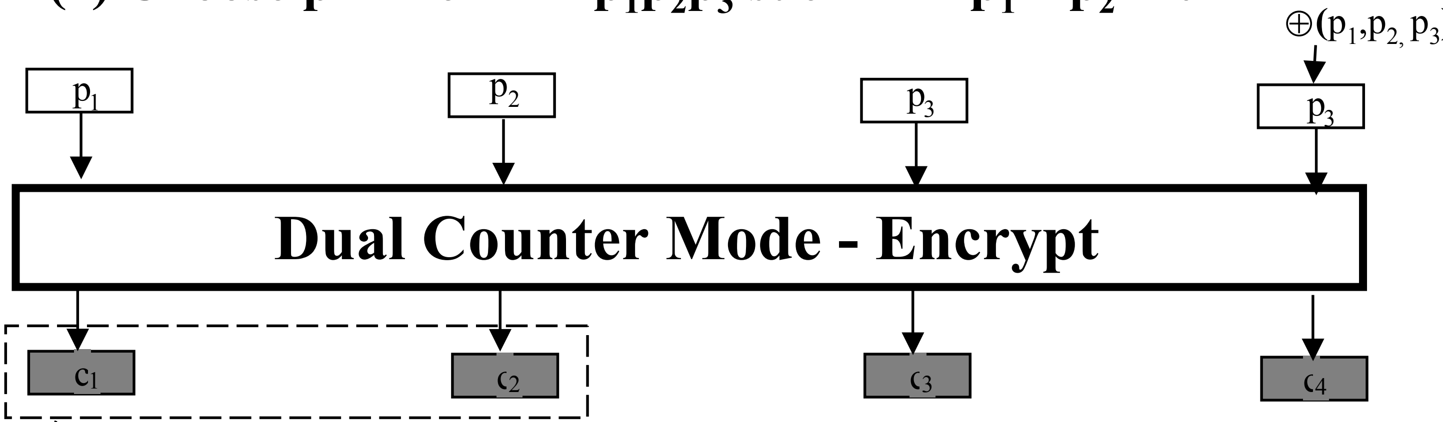
2. **Claim :** known **(f) LFSR** $\Rightarrow (x_0 \oplus x_j \Rightarrow x_0)$

   Find x_0 ; e.g., choose **plaintexts** $P = p_1 = 0$ and $P' = p_1 p_2 = 00$
   get **ciphertexts** $C = c_1\, c_2$ **and** $C' = c'_1 c'_2 c'_3$; note $x_0 \oplus x_2 = c_2 \oplus c'_2$

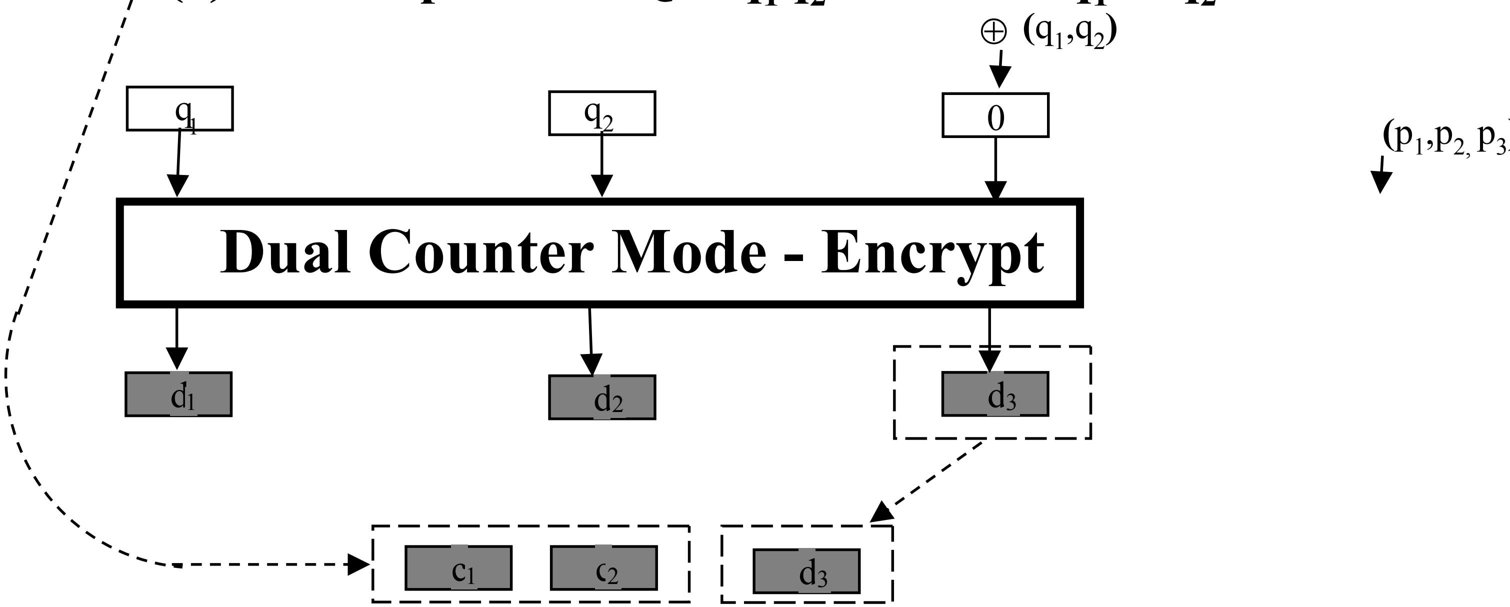
   Then construct a *valid forgery*; e.g., choose plaintext $P = p_1 p_2$ **such that** $p_1 = p_2$
   get **ciphertext** $C = c_1 c_2\, c_3$; then

   $C' = c_1 c'_2 \neq C$, where $c'_2 = c_2 \oplus x_0 \oplus x_2$ is a *valid forgery*

(1) Choose plaintext $P = p_1p_2p_3$ such that $p_1 \oplus p_2 = 0$



(2) Choose plaintext $Q = q_1q_2$ such that $q_1 = q_2 = 0$



(3) Forge ciphertext $C' = c_1c_2d_3$, which decrypts correctly

NSA's Dual Counter Mode - Version 2 (IPsec)

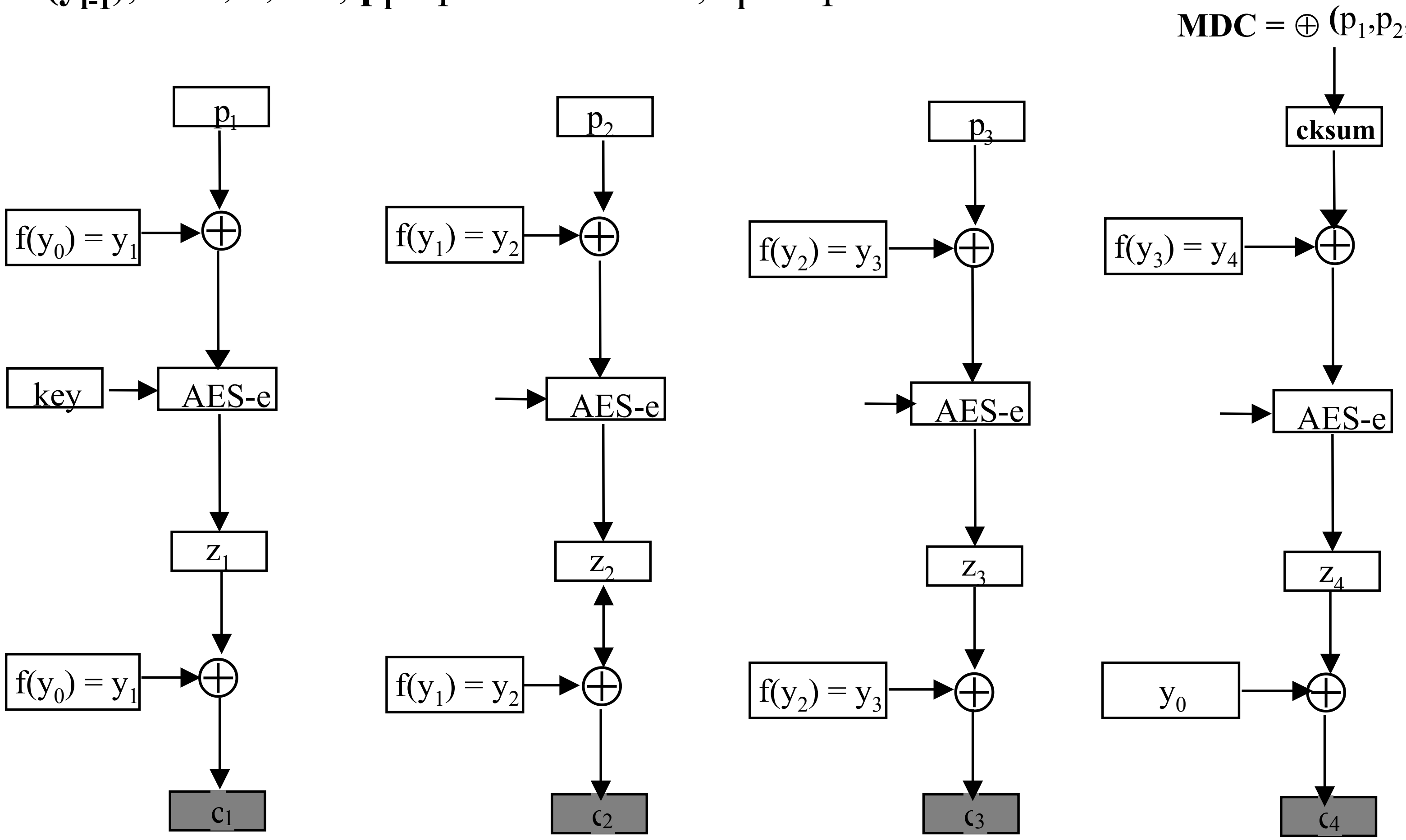
f = connection polynomial of degree W of a LFSR (W = width of block cipher)

$y^P_0 = x_0 \boxplus$ <SEQ^P SPI padding^P> for each message P,

where padding is the bit-wise complement of SEQ^P SPI

x_0 is not (cannot be) generated randomly per message => encryption is still not probabilistic

$y_i = f(y_{i-1})$, i = 1,..., n+1; p_i = plaintext block, **c_i** = ciphertext block



Attacks against the Dual Counter Mode - Version 2 (IPsec)

Secrecy and Integrity

1. *Fact* : The state update function of a (non-singular) LFSR (f) is linear.
   $\Rightarrow f(a \oplus b) = f(a) \oplus f(b)$

2. *Claim* : If an Adversary can force SEQ^P and SEQ^Q of a SPI such that
   $y^P_0 = y^Q_0 \oplus c$, where c is a known constant, then
   (a) secrecy and (b) integrity are broken

3. Example: find an SPI such that Probability $[y^P_0 = y^Q_0 \oplus c] = 1/8$

$$y^Q_0 = x_0 \boxplus \langle SEQ^Q\ SPI\ padding^Q \rangle = \langle 110\ldots0,\ SPI,\ 001\ldots1, \neg SPI \rangle \boxminus$$
$$y^P_0 = x_0 \boxplus \langle SEQ^P\ SPI\ padding^P \rangle = \langle 100\ldots0,\ SPI,\ 011\ldots1, \neg SPI \rangle$$

$$c = \langle 010\ldots0, 0\ldots0, 110\ldots0, 0\ldots0 \rangle$$

$y^Q_0 = y^P_0 \boxplus c \Rightarrow$ Probability $[y^Q_0 = y^P_0 \oplus c] = 1/8$

Examples of State Characteristics of a Mode

speed increases

Stateless

- needs good, secure source of randomness *per message*
- no state to maintain across messages (other than key)
- Execution: $\geq$ n+3 block cipher invocations;
- Latency: $\geq$2 block cipher invocations in parallel execution

Stateful Sender

- state (e.g., message counter) maintained by sender
- protection of sender state (e.g., counter integrity) across messages
- Execution: n+2 block cipher invocations
- Latency: 2 block cipher invocations in parallel execution

Stateful

- state : shared variables (other than key)
- protection of state secrecy, integrity across messages
- more susceptible to failures, intrusion
- Execution: *n+1* block cipher invocations -
- Latency: ≈ *1* block cipher invocation in parallel execution

robustness increases

XCBC Encryption

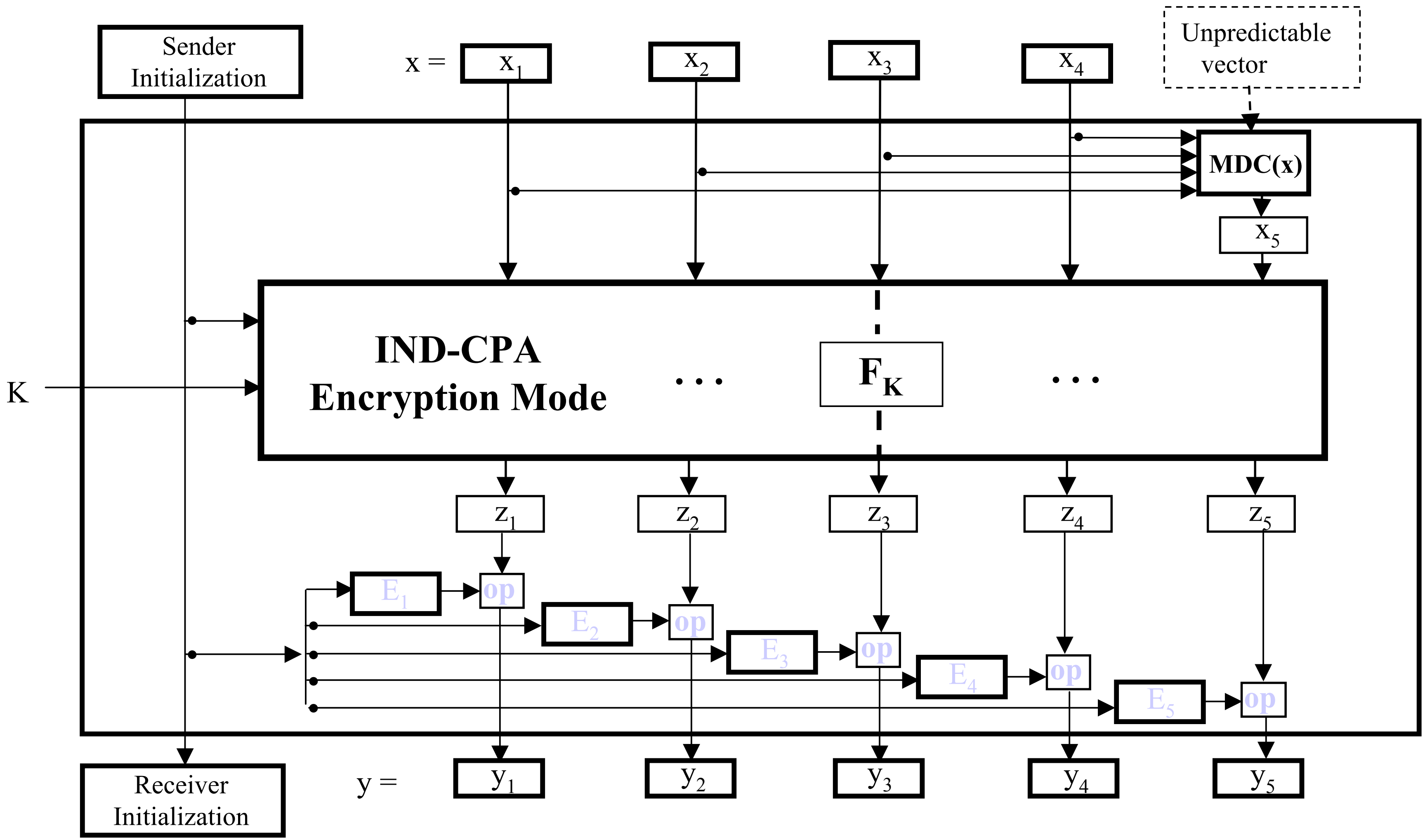
Fact: Encryption is not intended to provide integrity (authentication)

Motivation

- **Define family of encryption modes to help provide authenticated encryption using only non-cryptographic "redundancy" functions**

- **Security claims: IND-CPA confidentiality and EF-CPA integrity, reasonable bounds**

Example 1:
AE in 1 pass - 1 crypto primitive



XCBC-XOR [GD 00], IACBC [Jutla00]

Example 1:
AE in 1 pass - 1 crypto primitive

… Under What Conditions ?

1. IND-CPA encryption mode: processes block x_i, **$1 \leq i \leq n_m+1$,** and inputs result to block cipher (SPRP) F_K

2. **"op" has an inverse "op^{-1}"**

3. Elements E_i are **unpredictable**, **$1 \leq i \leq n_m+1$,** *and*
E^p_i **op^{-1}** E^q_j are **unpredictable,** where **$(p, i) \neq (q, j)$**
and messages p,q are encrypted with **same key K.**

4. Additional mechanisms for length control, padding

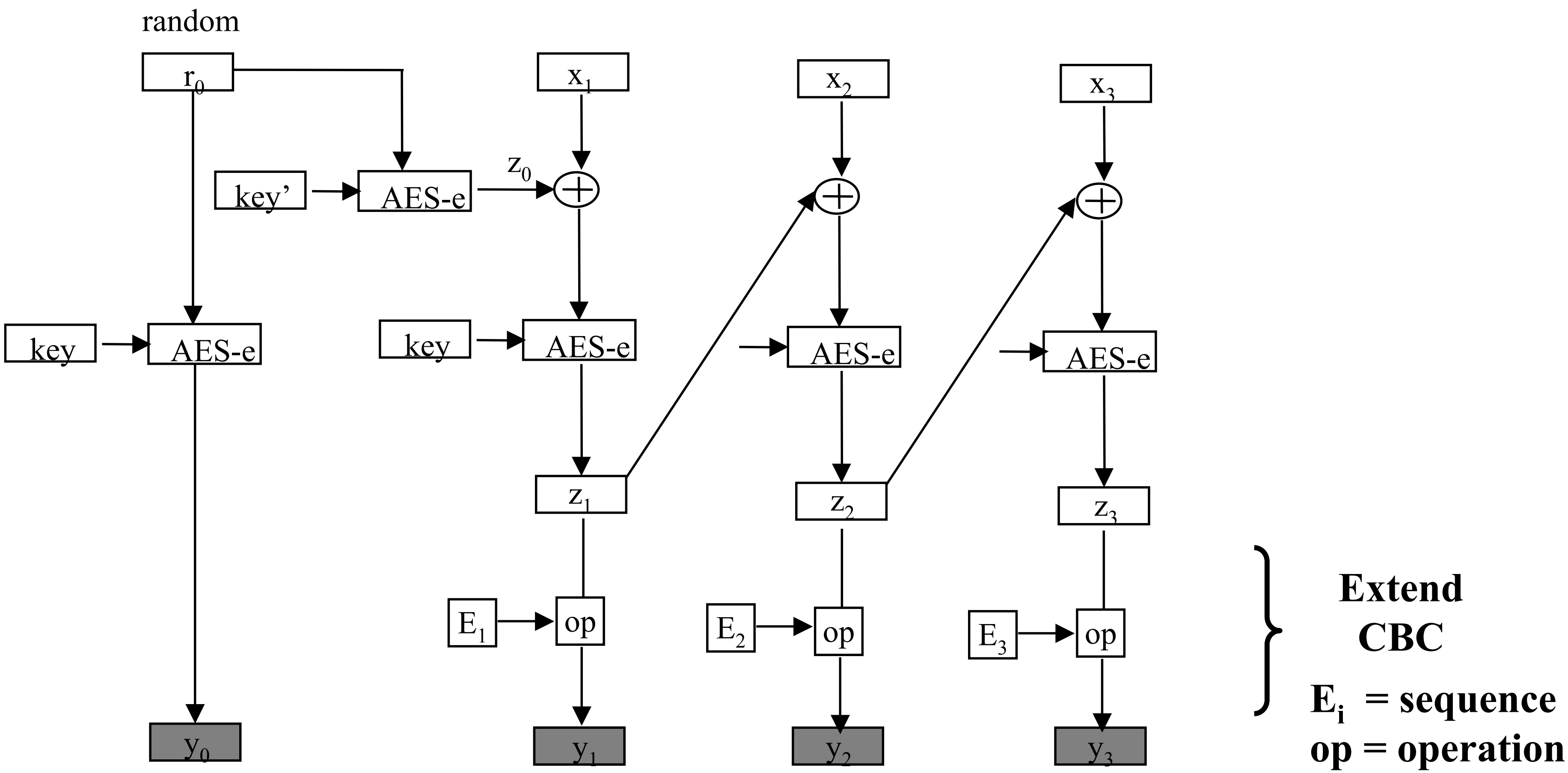
Examples

op = mod +/- ; $E_i = r_0 \times i$; $(E_0 = r_0 ; E_i = E_{i-1} + r_0)$ [GD00]
op = xor ; E_i = **pairwise (differential) independent** [Jutla00]
… and others [Rogaway01]

Optimal: n+1 cipher ops; latency in ||: 1 cipher op.

Stateless χCBC Scheme - Encryption of $x = x_1x_2x_3$

(single key is also possible)

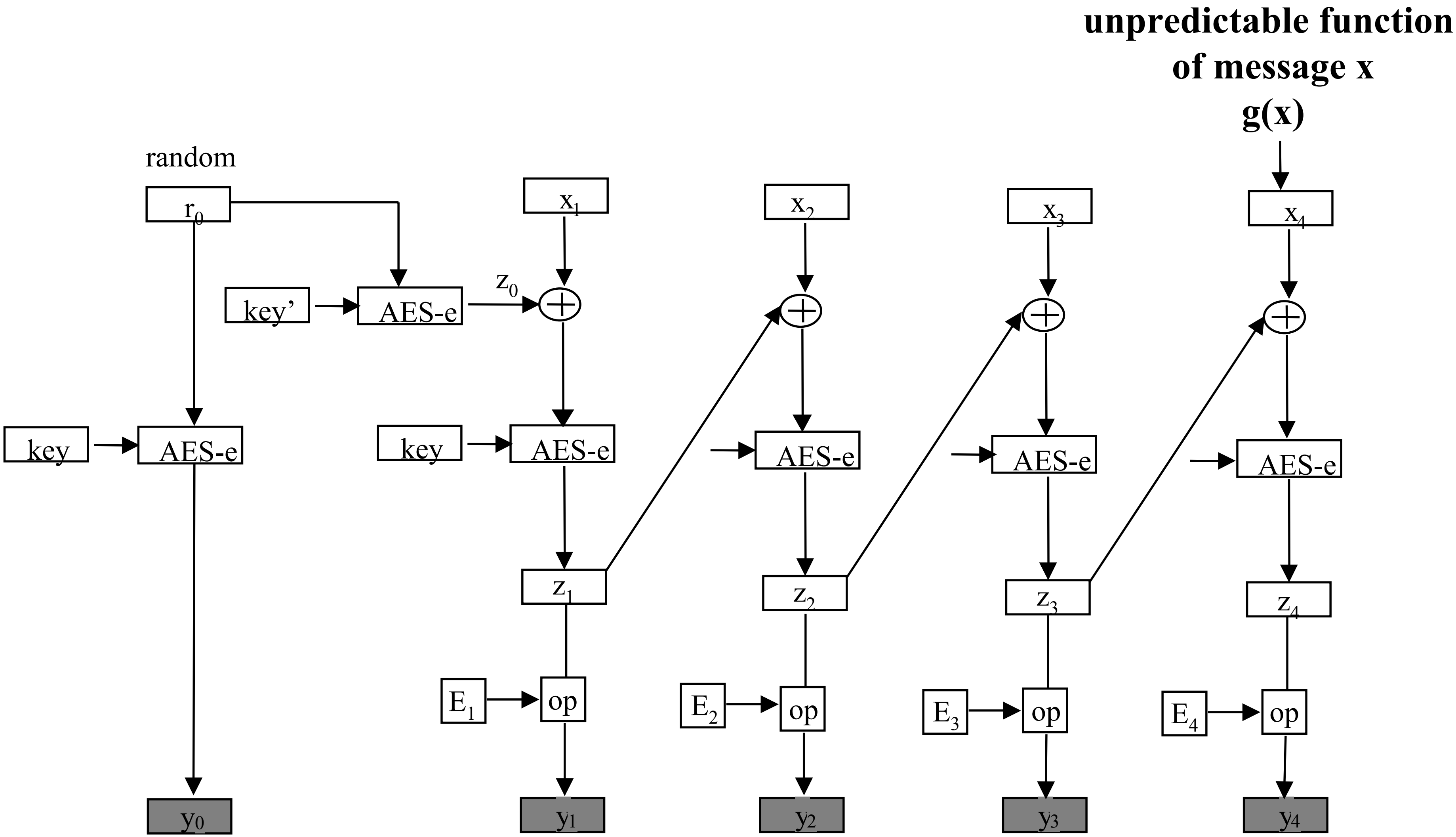


Examples of E_i and *op* combinations (+ is mod 2^l; $\oplus$ is bitwise exclusive-or)

op = + $E_i = E_{i-1} + r_0$, $E_0 = 0$ **(written as** $E_i = i \times r_0$**)**

Other S_i and *op* definitions exist (e.g., C.S. Jutla's and P. Rogaway's proposals)

Stateless XCBC-XOR Scheme - Encryption of $x = x_1x_2x_3$

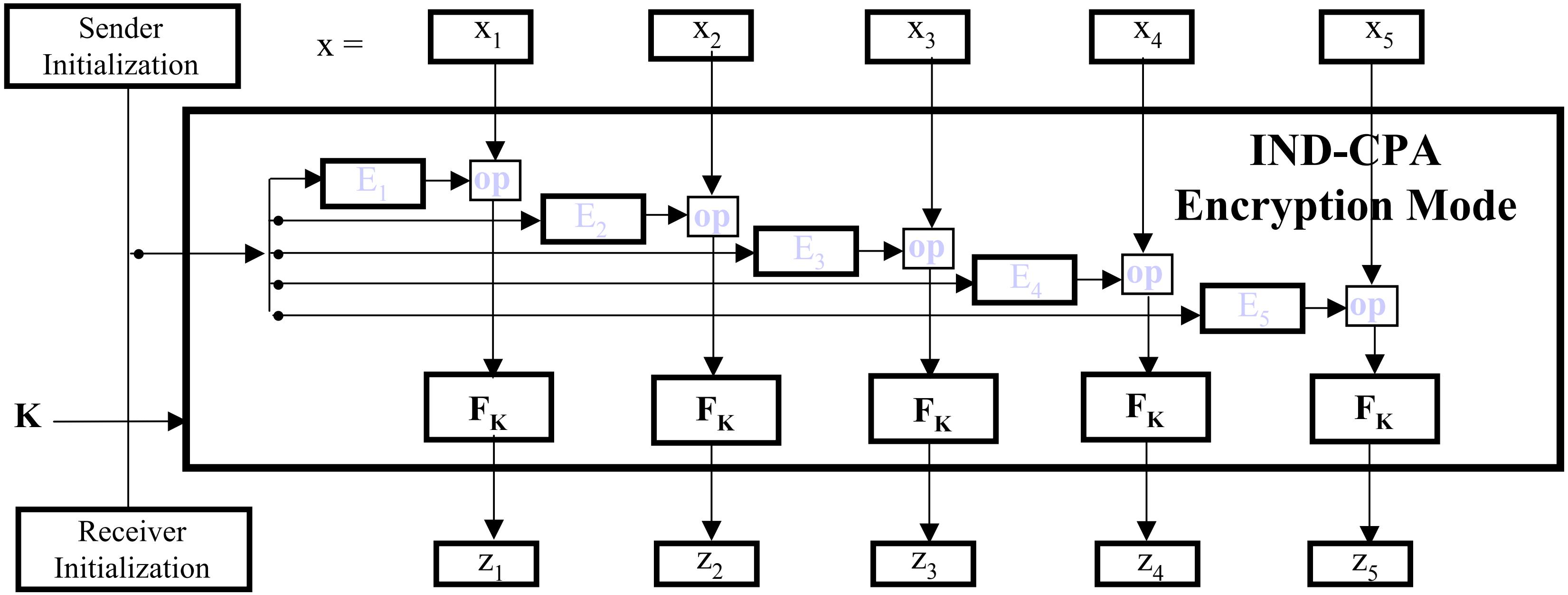


Example: $g(x) = x_1 \oplus x2 \oplus x3 \oplus z_0$;

Other examples of g(x) exist

Example 1:
AE in 1 pass - 1 crypto primitive

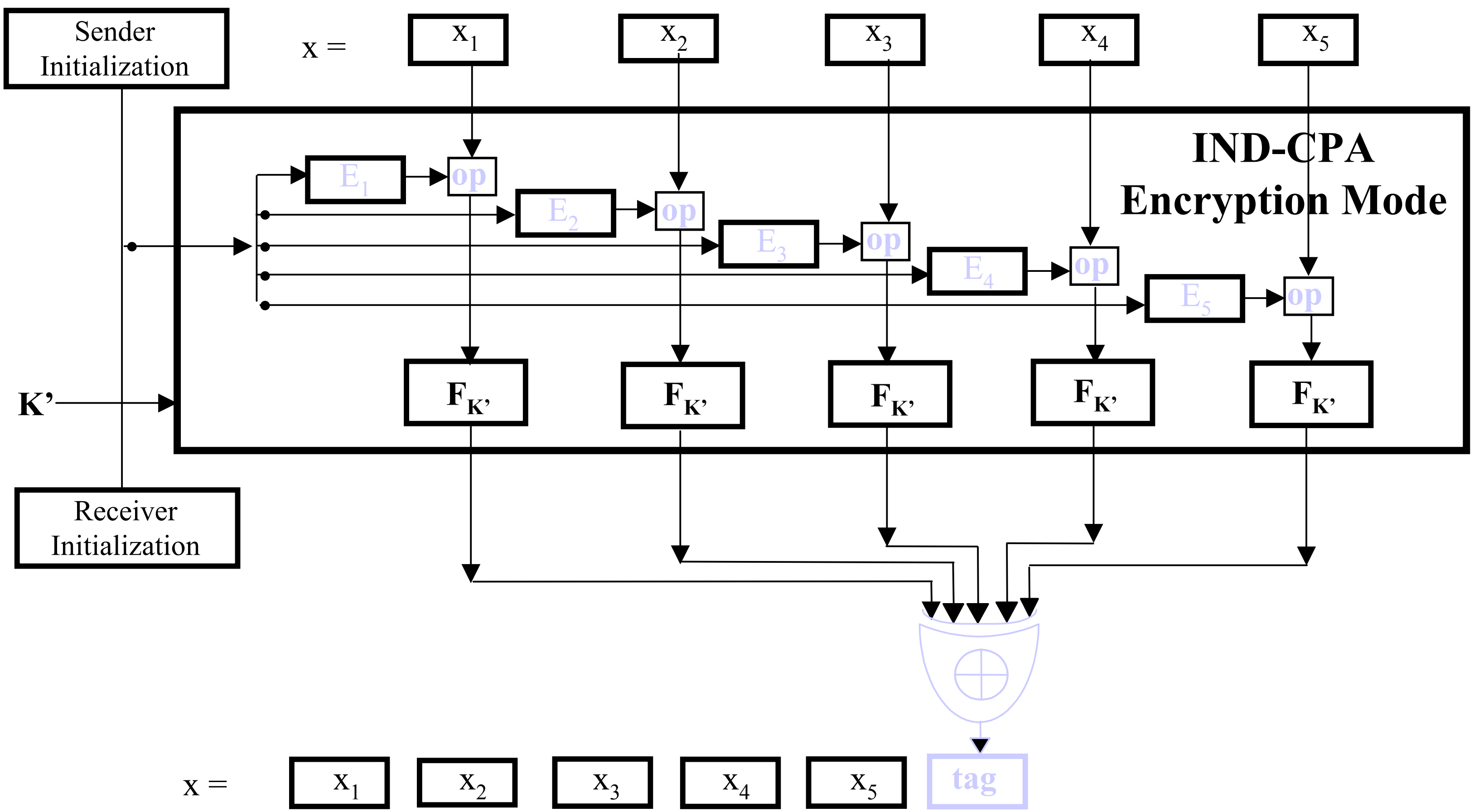
Same hardware used on input (viz., IAPM [Jutla00], XECB-XOR [GD00])



…. minimizes hardware footprint, and provides IND-CPA security and …

Example 1:
AE in 1 pass - 1 crypto primitive

… a (parallel) MAC w/ an extra XOR gate (viz., [G98, GD00])



Parallel Mode

Motivation

- Fully Parallel Mode like C.S. Jutla's IAPM using a different S_i
(S_i elements are *not* pairwise independent)

- Define family of parallel encryption modes to help provide integrity
with non-cryptographic "redundancy" functions

- Security Claims (w/ proof) : IND-CPA confidentiality and EF-CPA integrity,
reasonable bounds

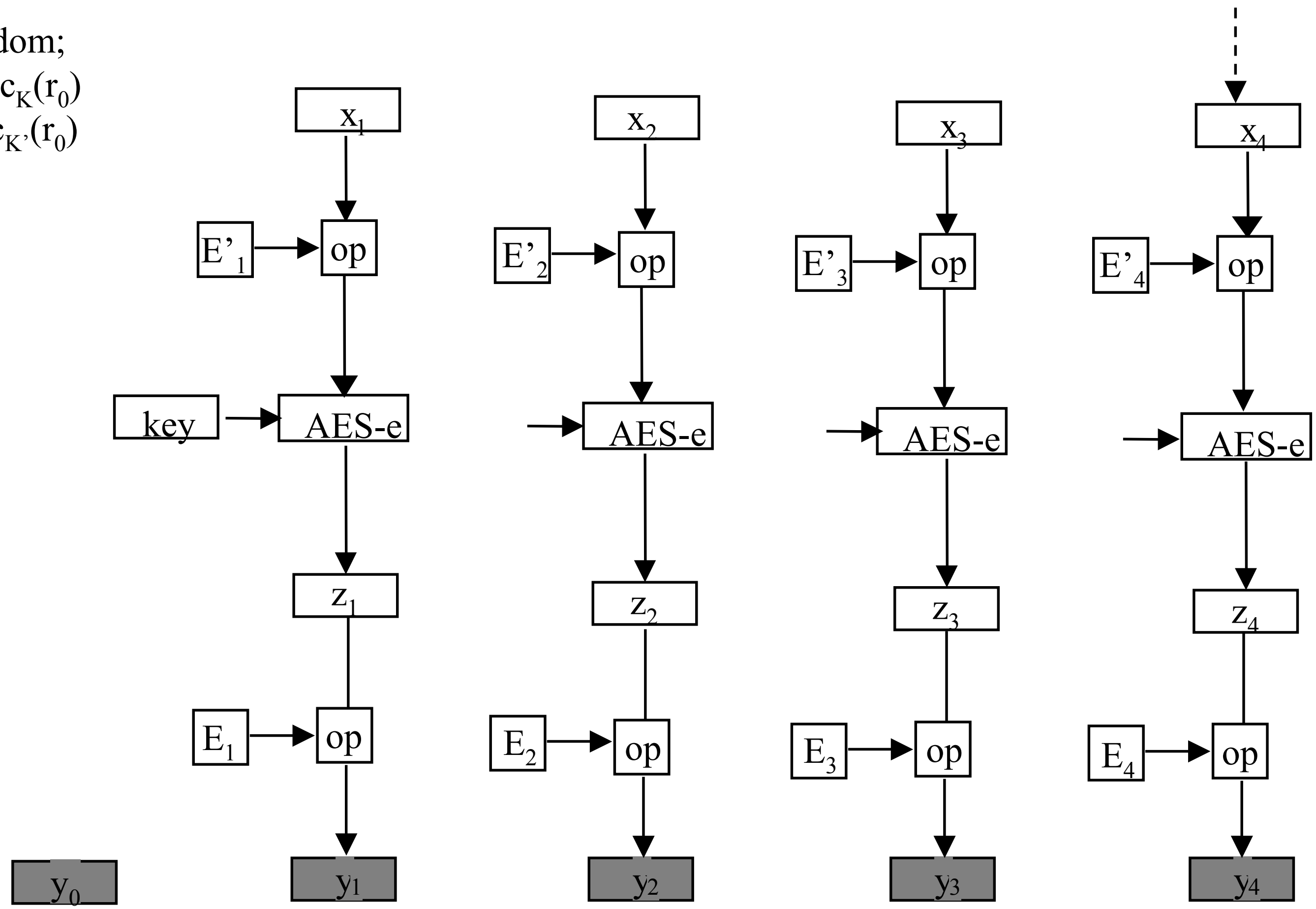
Stateless Parallel Mode - Encryption of $x = x_1x_2x_3$

(single key mode is also possible)

unpredictable function of message x

g(x)

r_0= random;
$y_0 = Enc_K(r_0)$
$z_0 = Enc_{K'}(r_0)$



Example: $g(x) = x_1 \oplus x2 \oplus x3 \oplus z_0$;

$y_i = Enc_K(x_i + E_i) + E_i$; $E_i = i \times r_0$;

Other examples of E_i, g(x) exist (e.g., C.S. Jutla's and P. Rogaway's proposals)

Three Distinct AE Modes of Operation

and other Candidates (NIST AES Modes of Operation Workshop)
October 20, 2000 and August 24, 2001

1. ***If CBC is retained as a standard AES mode, then the authenticated encryption mode is***
- **XCBC-XOR (January 31, 2000)**
- **plus interleaved parallel mode**

2. ***Parallel authenticated encryption modes (single confidentiality and integrity key)***
- **IAPM (April 14, 2000)**
- **XECB-XOR (August 24, 2000)**
- **OCB (September 2000 - February 2001)**

3. ***High-End (separate or independent key for confidentiality and integrity modes)***
- **ctr-mode for encryption (already selected)**
- **XECB-MAC (March 31, 2000), PMAC (Sept. 2000 - Feb. 2001) for integrity**

Status: No Authenticated Encryption Mode Selected by NIST for AES (so far)
Possible reason: Intellectual Property claims (viz., dates of inventions above)