COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2020

Announcements/Reminders

HW2 due September 29

- Submit through Gradescope

PR1 Due October 6

Previously on COS 433…

Pseudorandom Functions

Functions that "look like" random functions

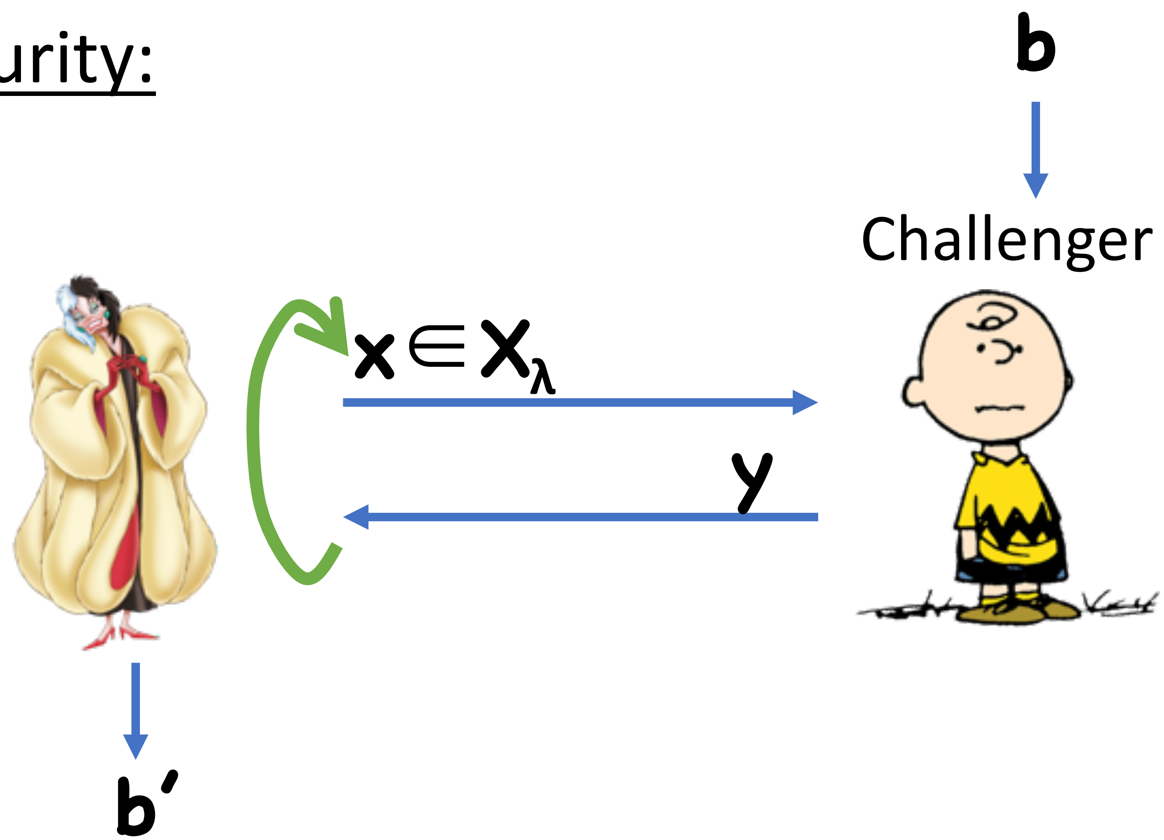
Syntax:

- Key space $\mathbf{K}_\lambda$
- Domain $\mathbf{X}_\lambda$
- Co-domain/range $\mathbf{Y}_\lambda$
- Function $\mathbf{F}:\mathbf{K}_\lambda \times \mathbf{X}_\lambda \rightarrow \mathbf{Y}_\lambda$

Correctness: $\mathbf{F}$ is a function (deterministic)

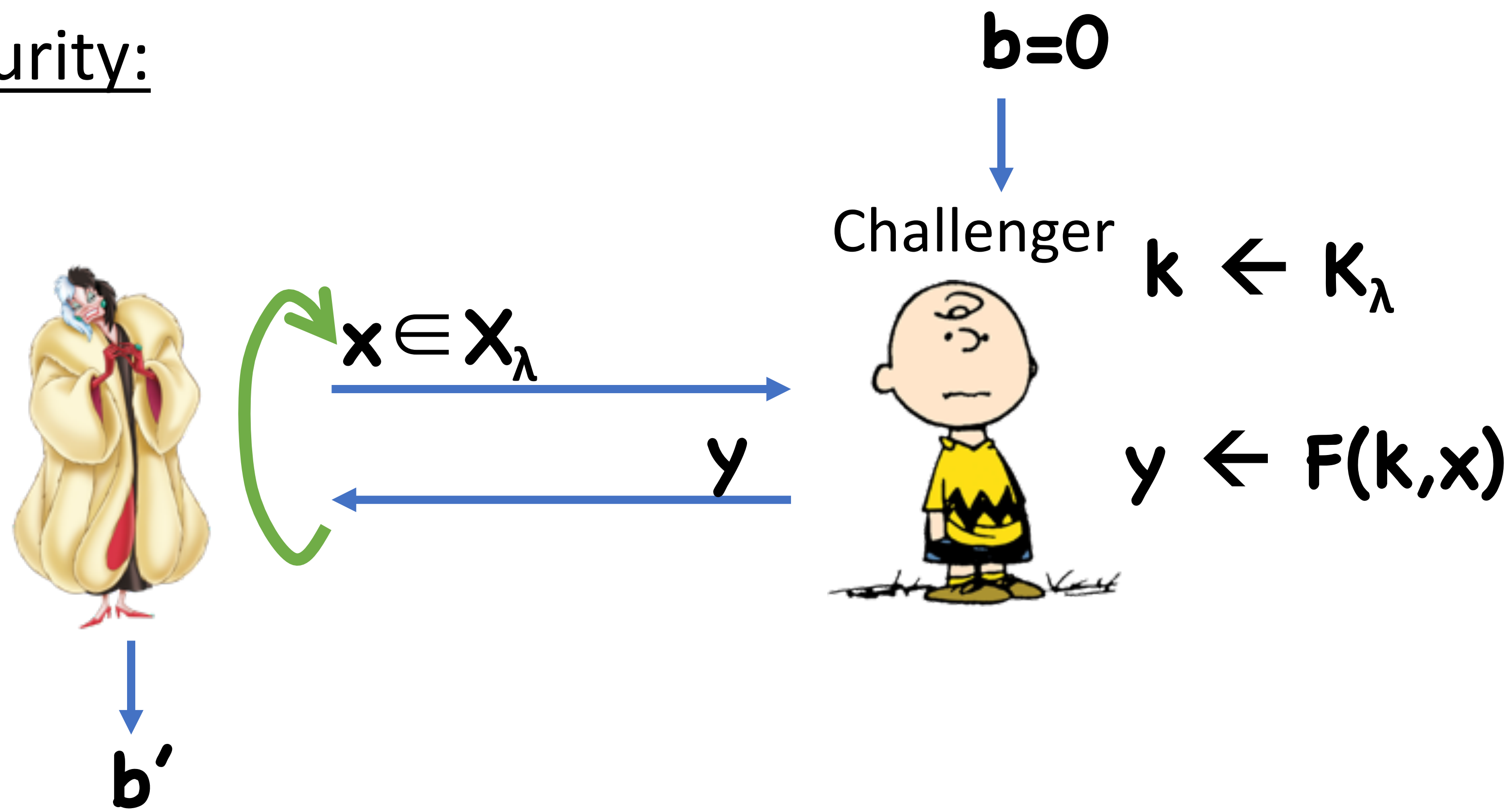
Pseudorandom Functions

Security:



Pseudorandom Functions

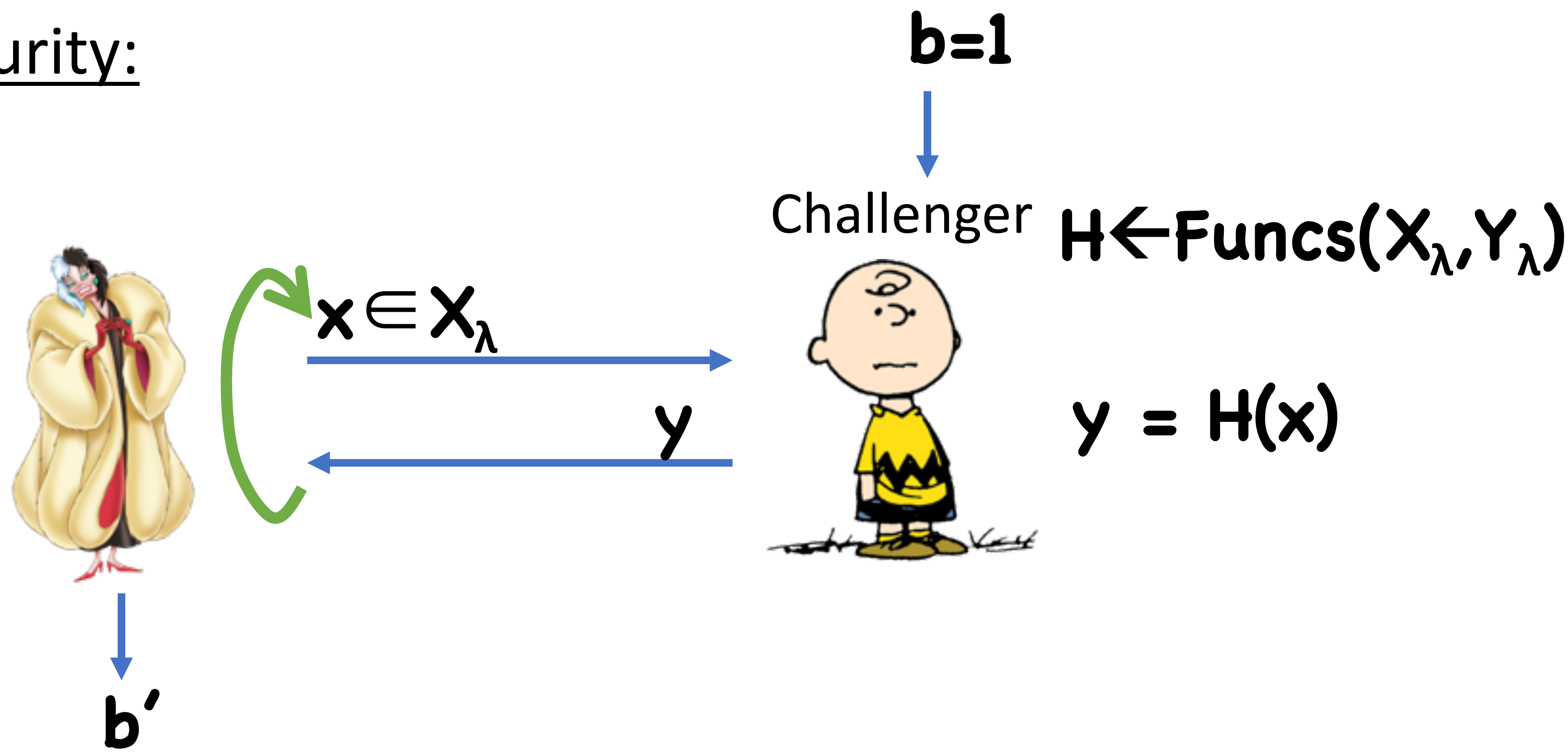
Security:



PRF-Exp$_0$( , λ)

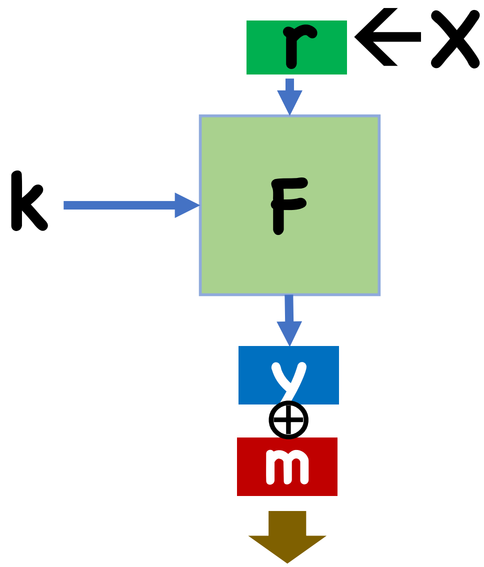
Pseudorandom Functions

Security:



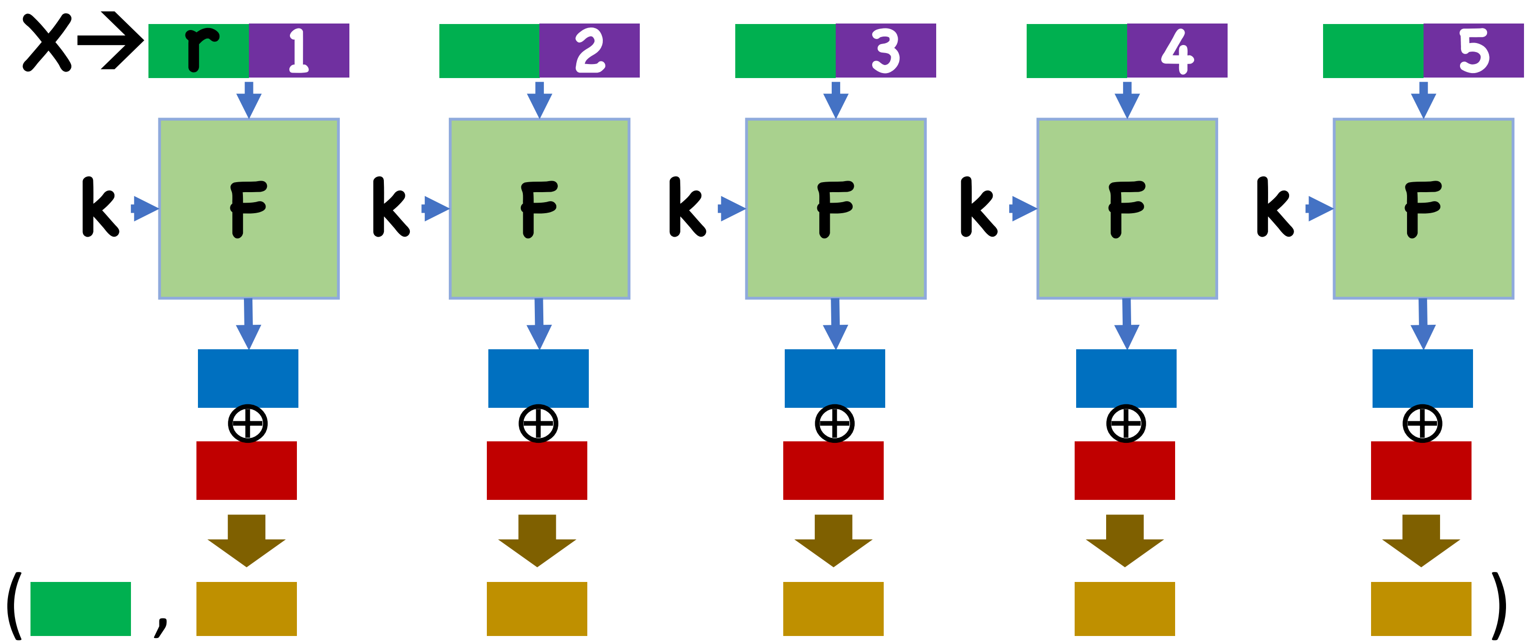
$\mathbf{PRF\text{-}Exp_1(}$  $\mathbf{, \lambda)}$

Using PRFs to Build Encryption



Ciphertext = ( , c)

Counter Mode



Today

Block ciphers, more modes of operation

Begin constructing block ciphers/PRFs

Pseudorandom Permutations

(also known as block ciphers)

Functions that "look like" random **permutations**

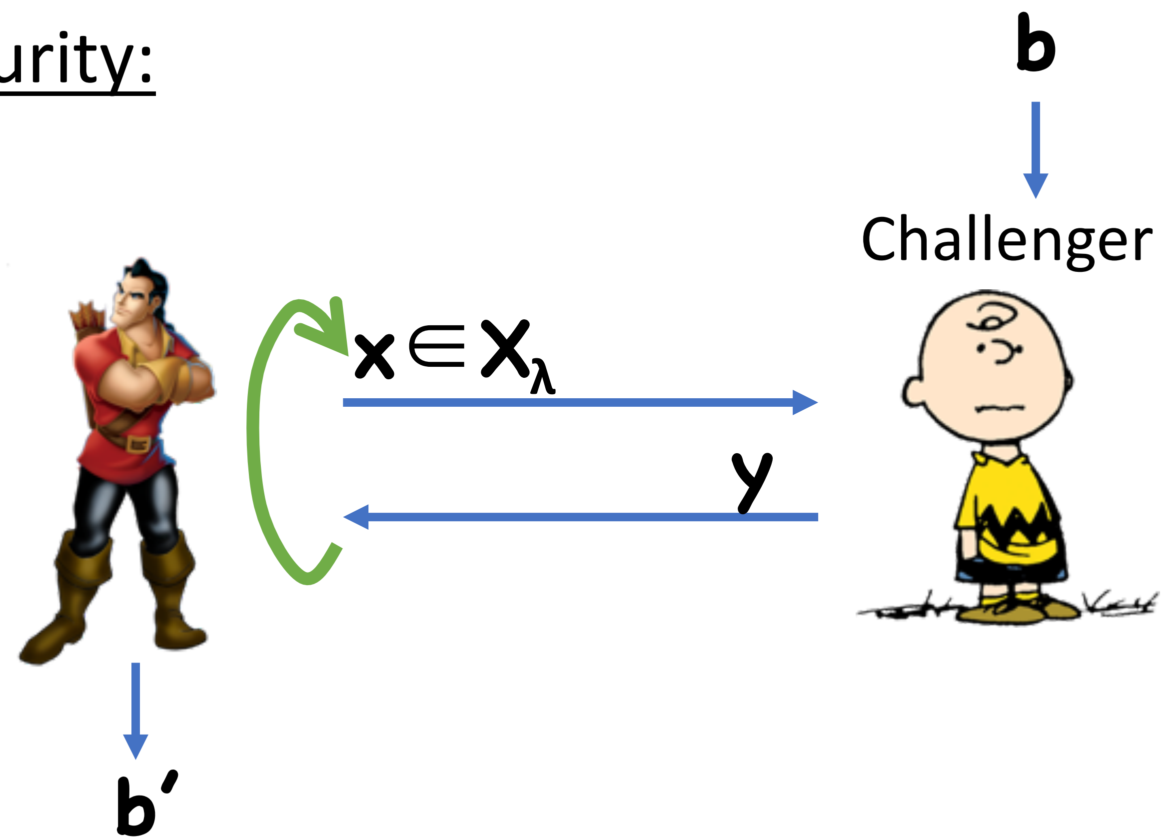
Syntax:

- Key space $\mathbf{K}_\lambda$
- Domain=Range= $\mathbf{X}_\lambda$
- Function $\mathbf{F}:\mathbf{K}_\lambda \times \mathbf{X}_\lambda \rightarrow \mathbf{X}_\lambda$
- Function $\mathbf{F}^{-1}:\mathbf{K}_\lambda \times \mathbf{X}_\lambda \rightarrow \mathbf{X}_\lambda$

Correctness: $\forall \mathbf{k},\mathbf{x},\ \mathbf{F}^{-1}(\mathbf{k},\ \mathbf{F}(\mathbf{k},\ \mathbf{x})\ ) = \mathbf{x}$

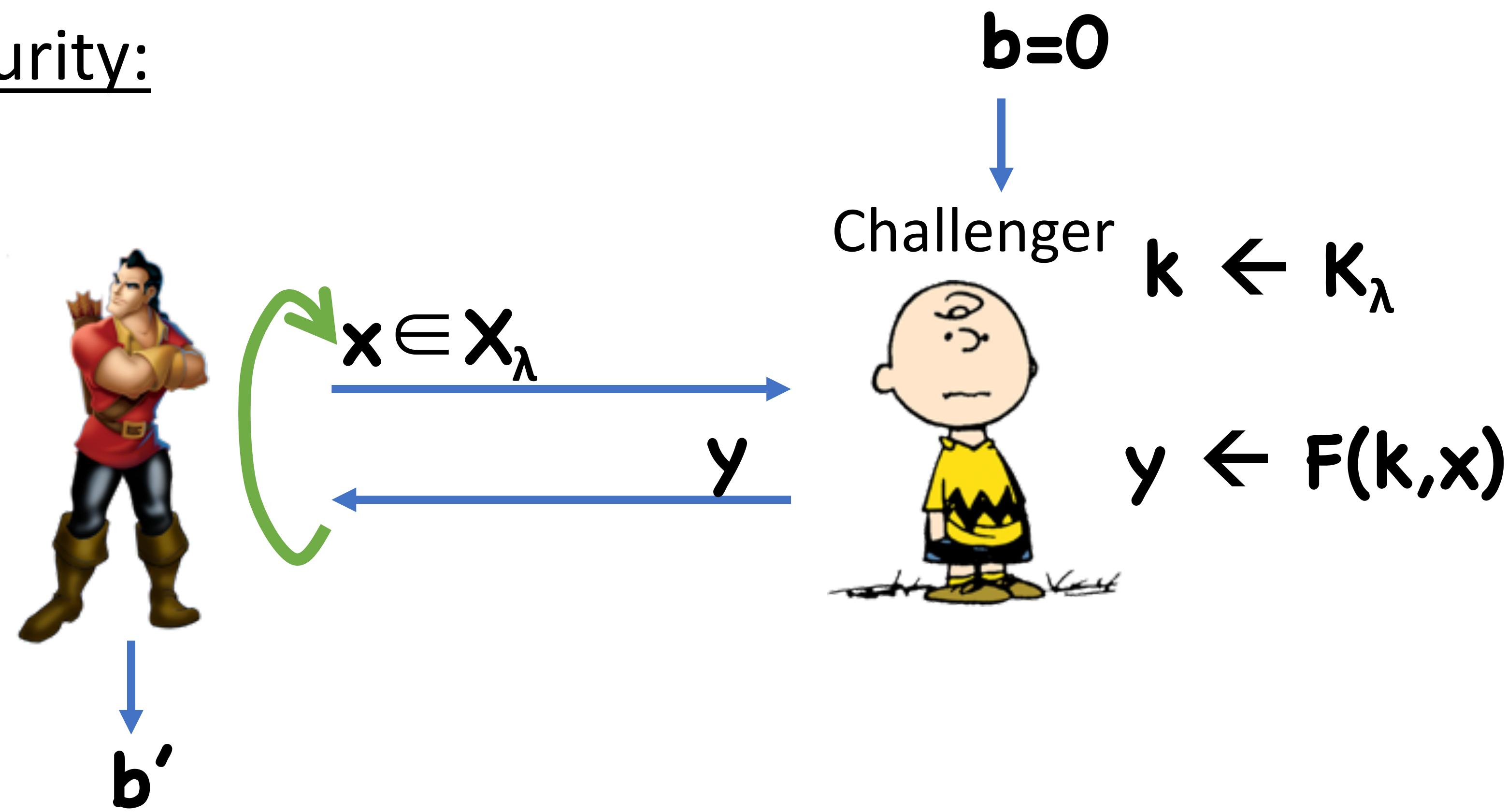
Pseudorandom Permutations

<u>Security:</u>



Pseudorandom Permutations

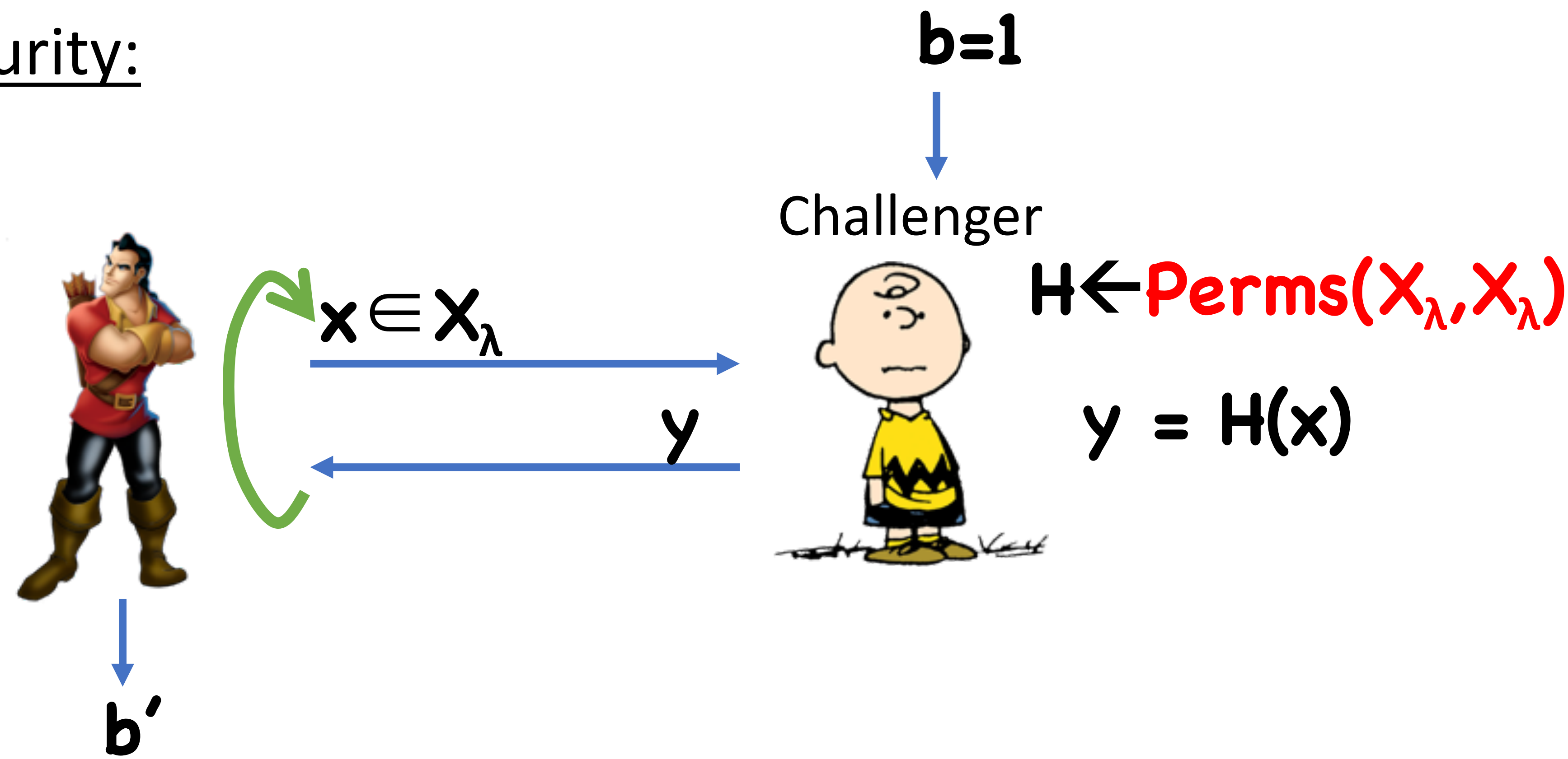
Security:



PRF-Exp$_0$(, λ)

Pseudorandom Permutations

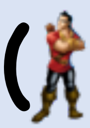
<u>Security:</u>

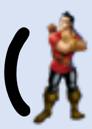


PRF-Exp$_1$( **, λ)**

PRP Security Definition

Definition: F is a secure PRP if, for all  running in polynomial time, $\exists$ negligible **ε** such that:

$\Big|$ Pr[1←PRF-Exp$_0$( , λ)]

− Pr[1←PRF-Exp$_1$( , λ)] $\Big| \leq \varepsilon(\lambda)$

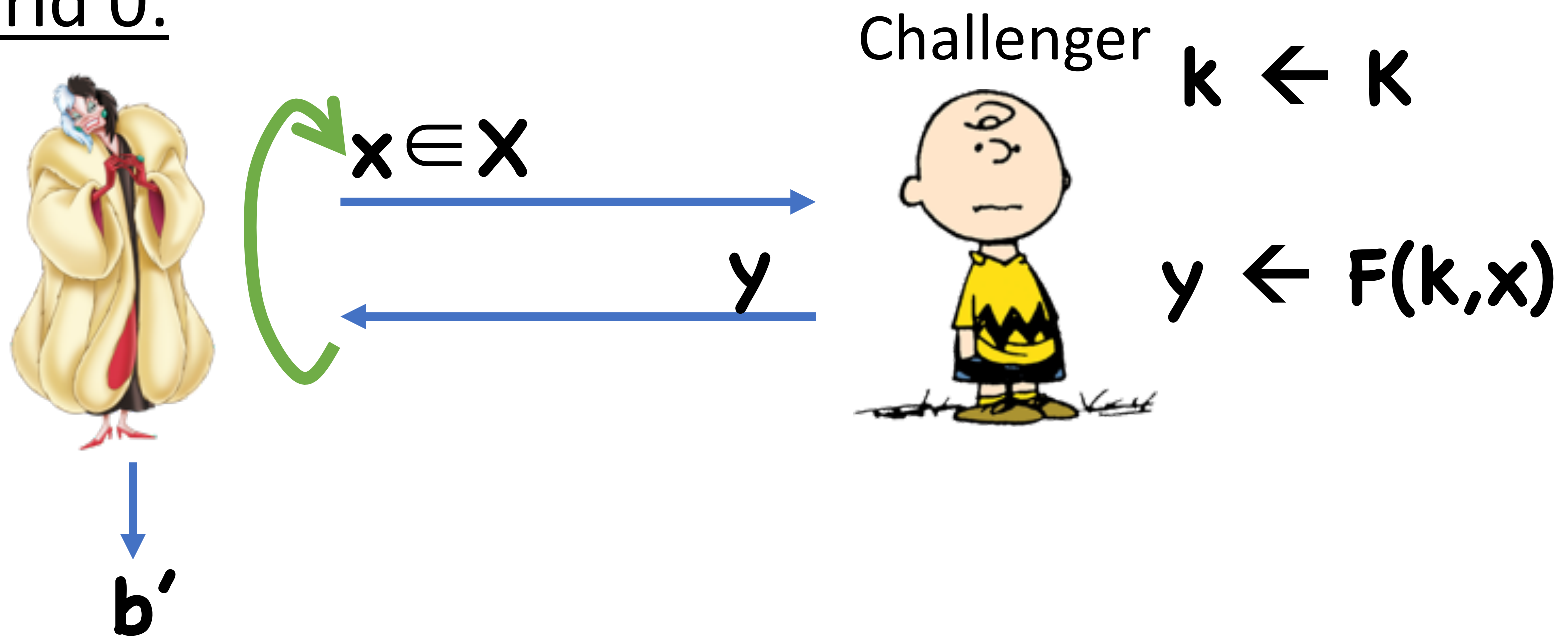
Theorem: Assuming $|X_\lambda|$ is super-polynomial, a PRP (F, F^{-1}) is secure iff F is secure as a PRF

Proof

Secure as PRP $\Rightarrow$ Secure as PRF

- Assume , hybrids

Hybrid 0:

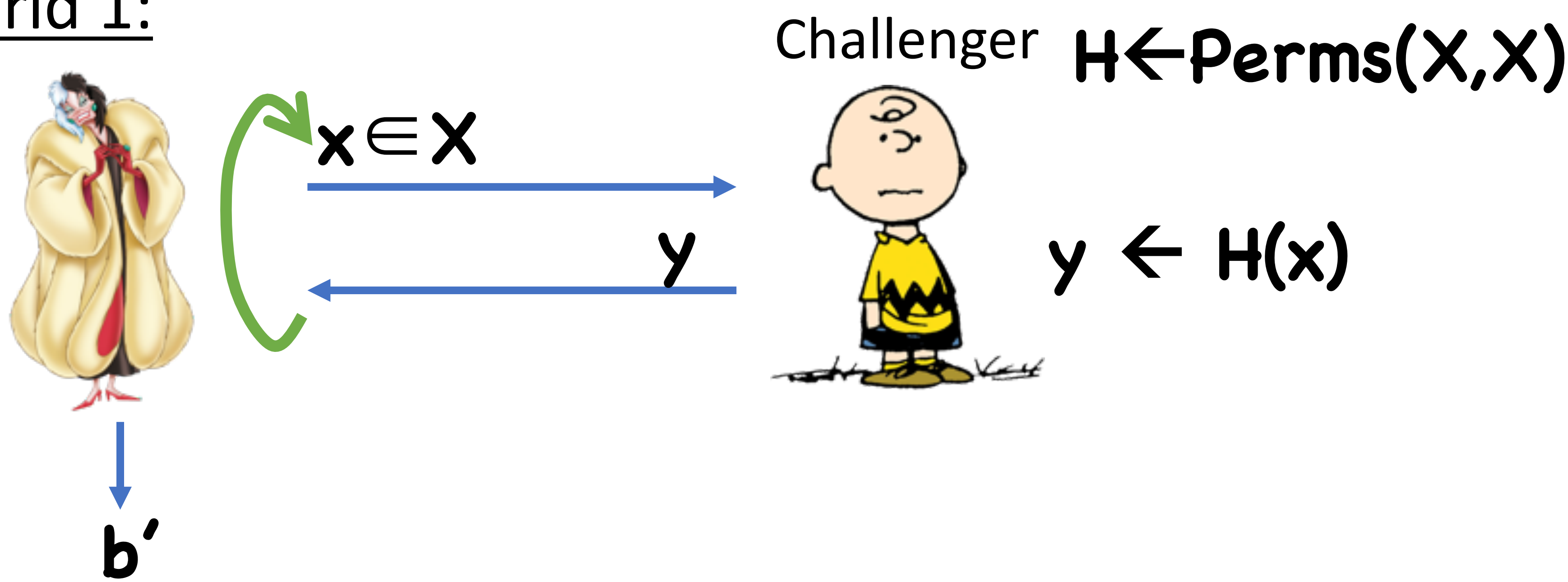


Proof

Secure as PRP ⇒ Secure as PRF

- Assume , hybrids

<u>Hybrid 1:</u>

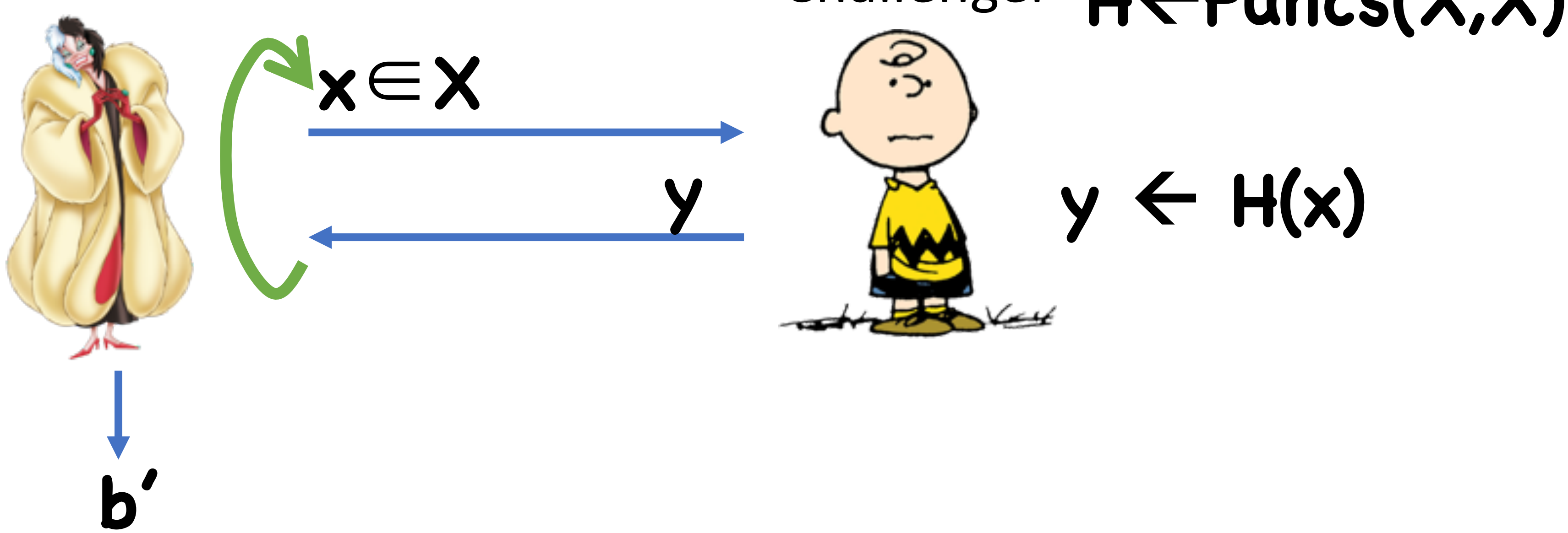


Proof

Secure as PRP $\Rightarrow$ Secure as PRF

- Assume , hybrids

<u>Hybrid 2:</u>



Proof

Secure as PRP $\Rightarrow$ Secure as PRF

- Assume , hybrids

Hybrids 0 and 1 are indistinguishable by PRP security

Hybrids 1 and 2?

- In Hybrid 1,  sees random **distinct** answers
- In Hybrid 2,  sees random answers
- Except with probability $\approx \mathbf{q^2/2|X_\lambda|}$, random answers will be distinct anyway

Proof

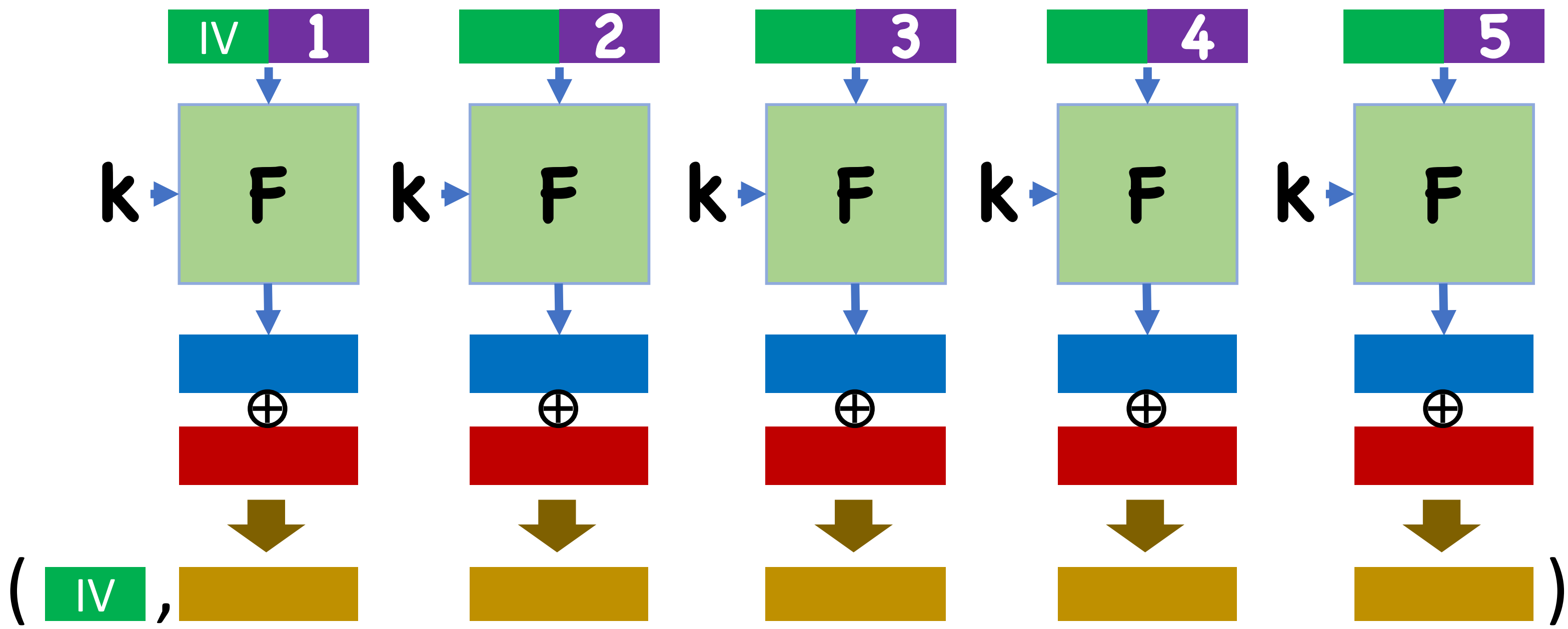
Secure as PRF $\Rightarrow$ Secure as PRP

- Assume , hybrids

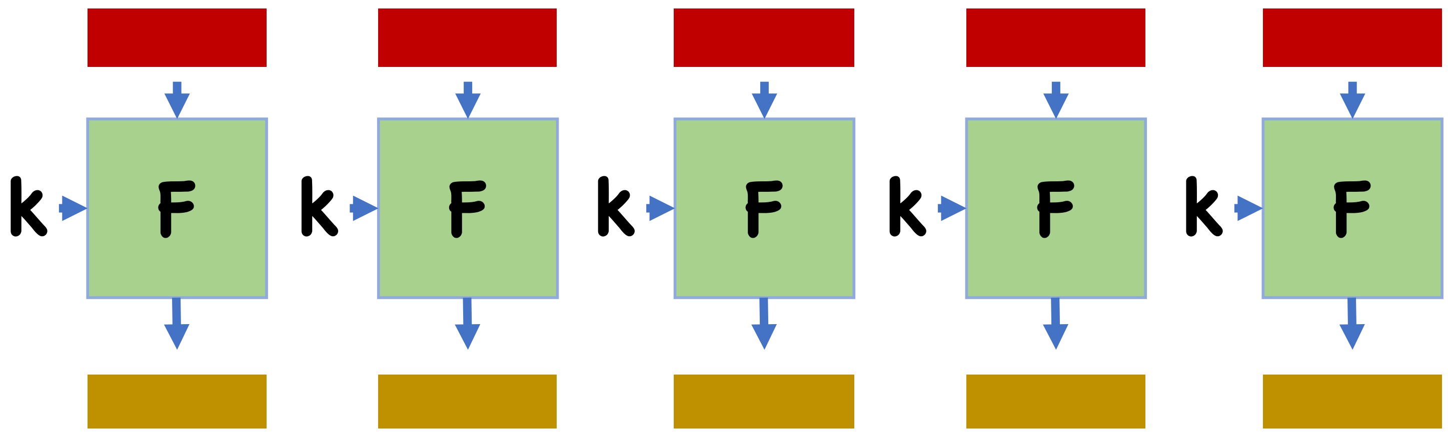
Proof essentially identical to other direction

How to use block ciphers for encryption

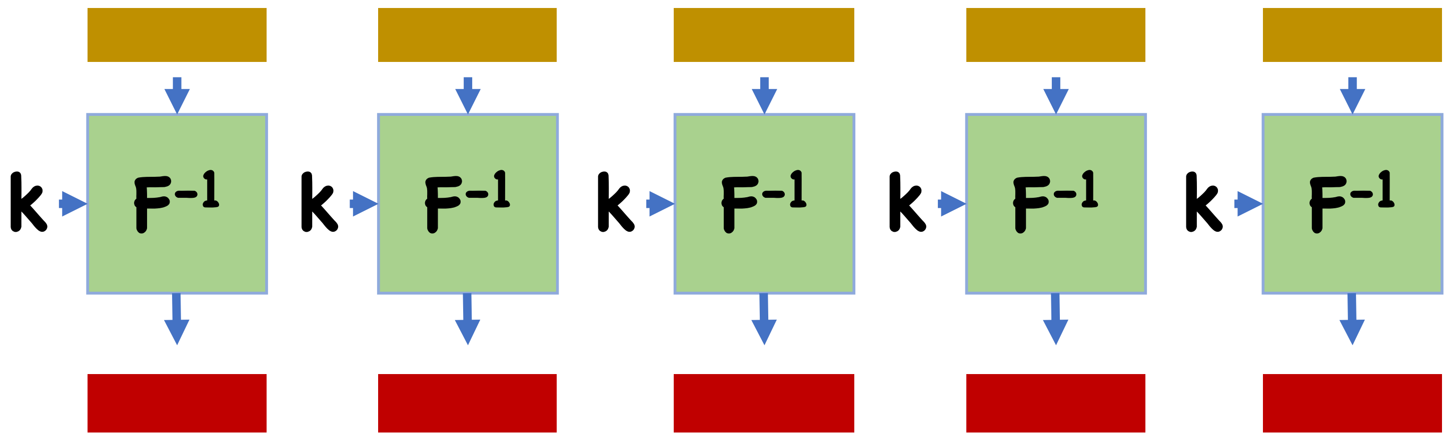
Counter Mode (CTR)



Electronic Code Book (ECB)



ECB Decryption



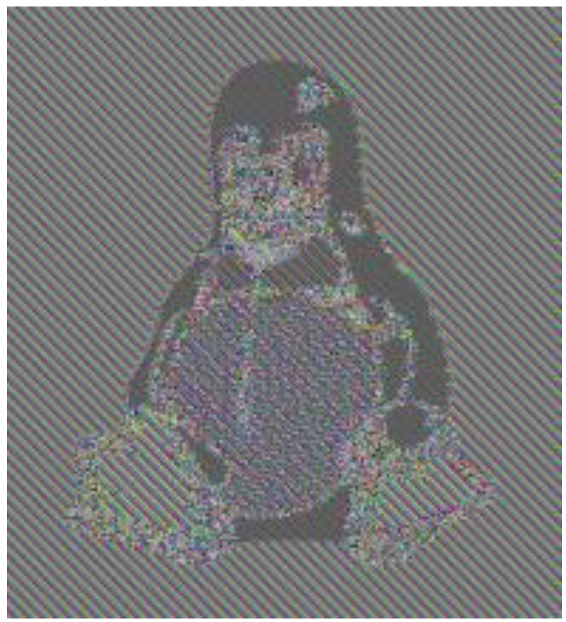
Security of ECB?

Is ECB mode CPA secure?

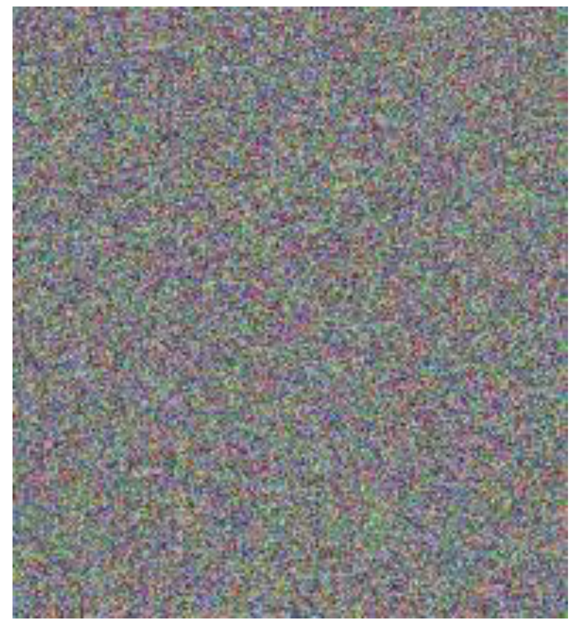
Is ECB mode *one-time* secure?

Security of ECB

Plaintex

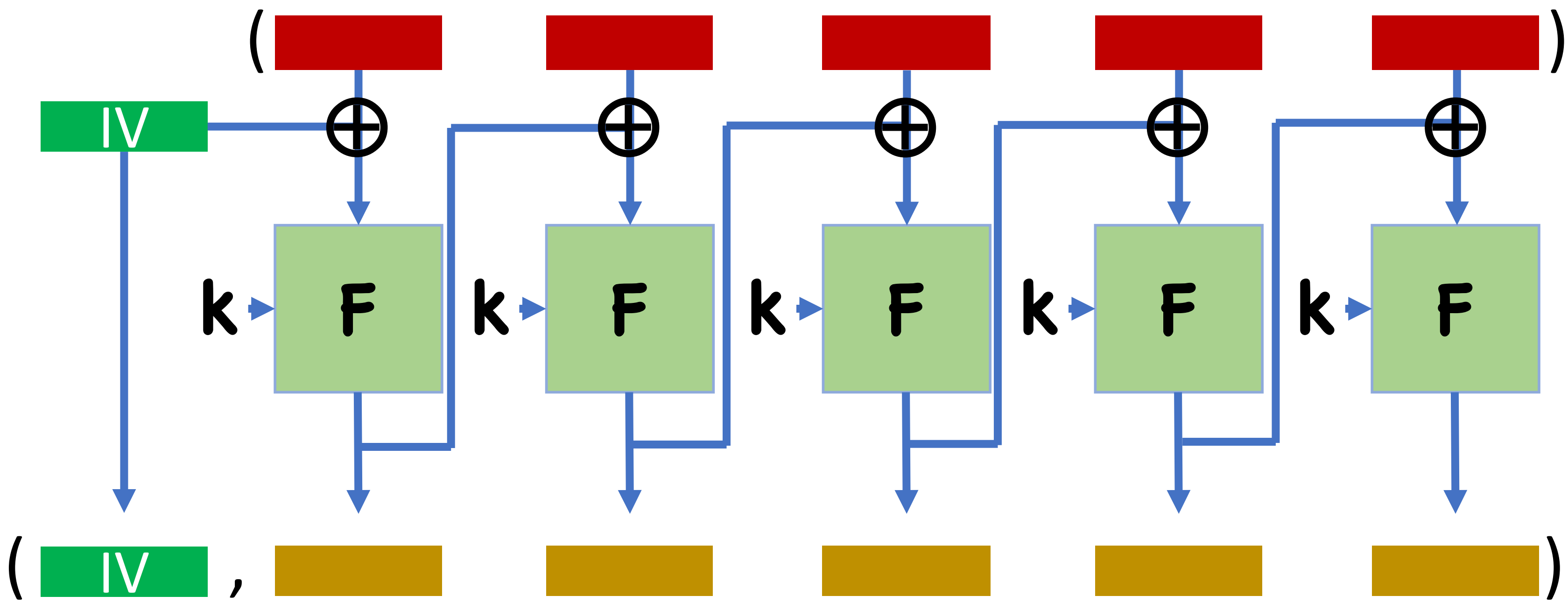


Ciphertext



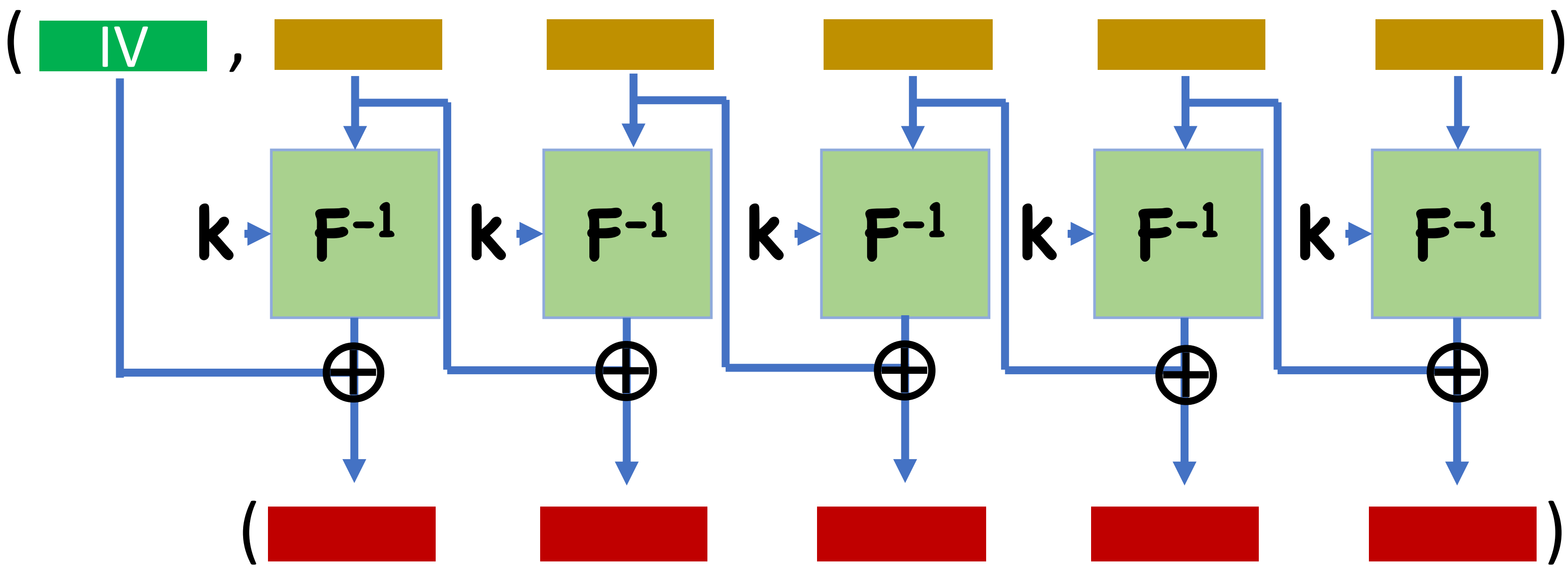
Ideal

Cipher Block Chaining (CBC) Mode



(For now, assume all messages are multiples of the block length)

CBC Mode Decryption



Theorem: If (F, F^{-1}) is a secure pseudorandom permutation and $|X_\lambda|$ is super-polynomial, then CBC mode encryption is CPA secure.

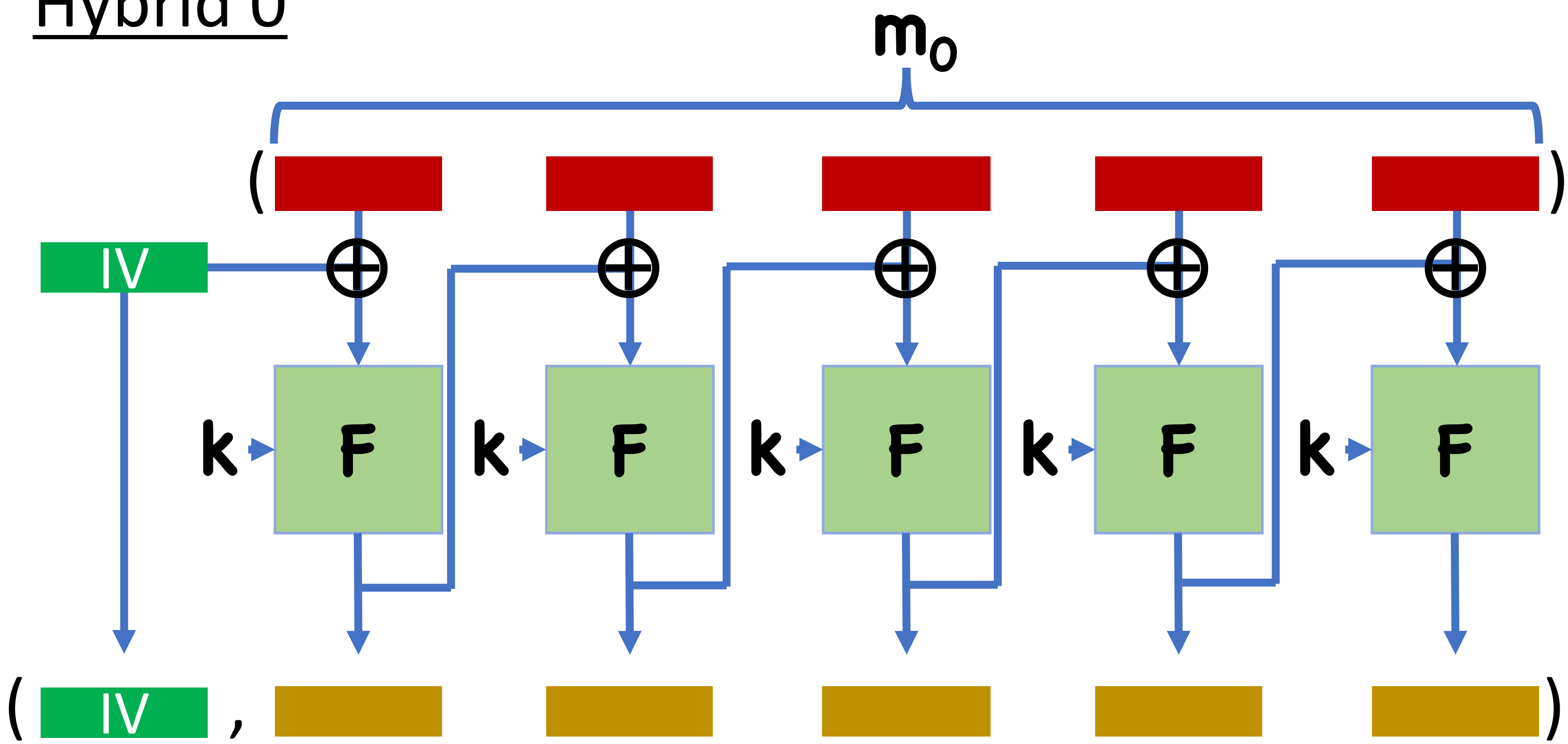
Proof Sketch

Assume toward contradiction an adversary  for CBC mode

Hybrids…

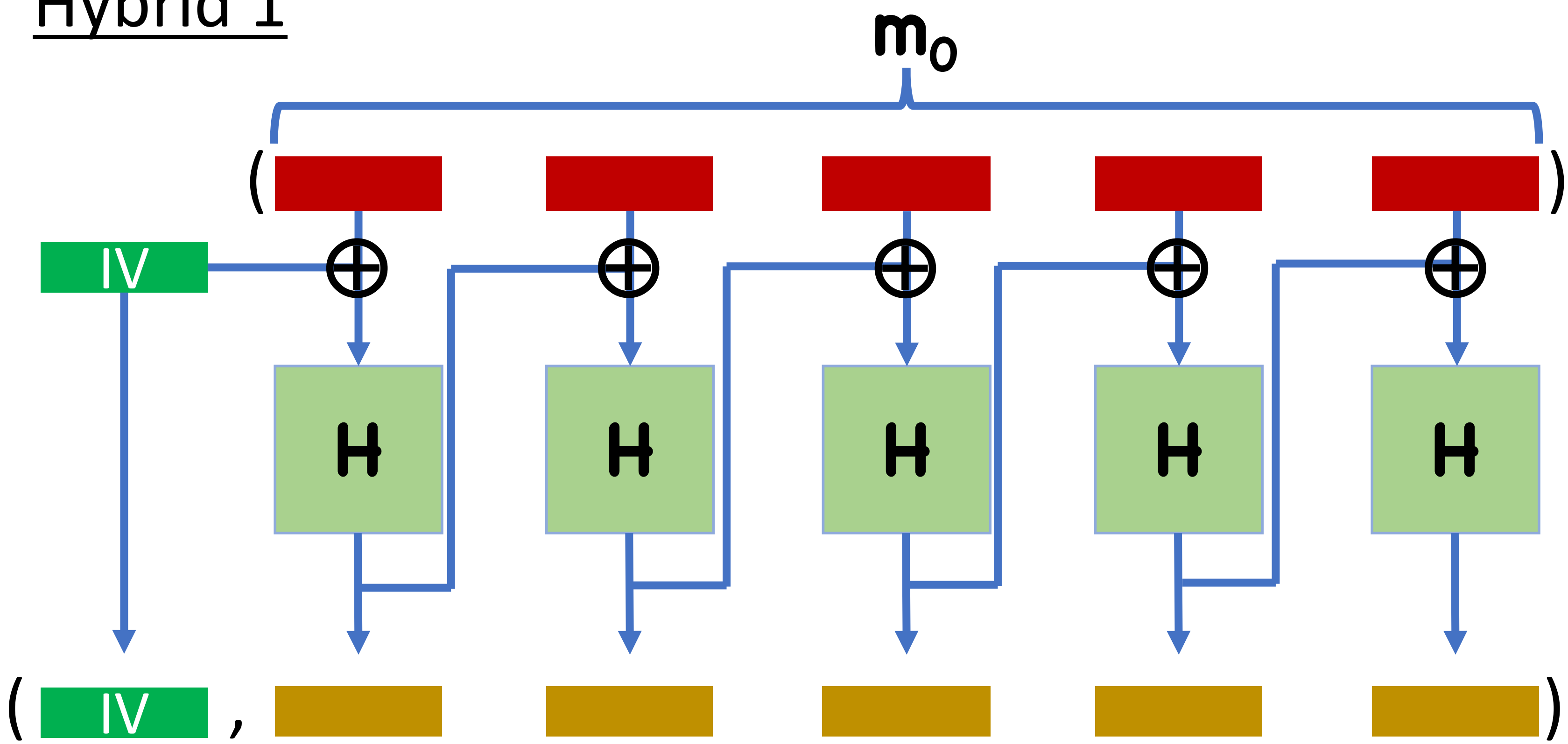
Proof Sketch

Hybrid 0



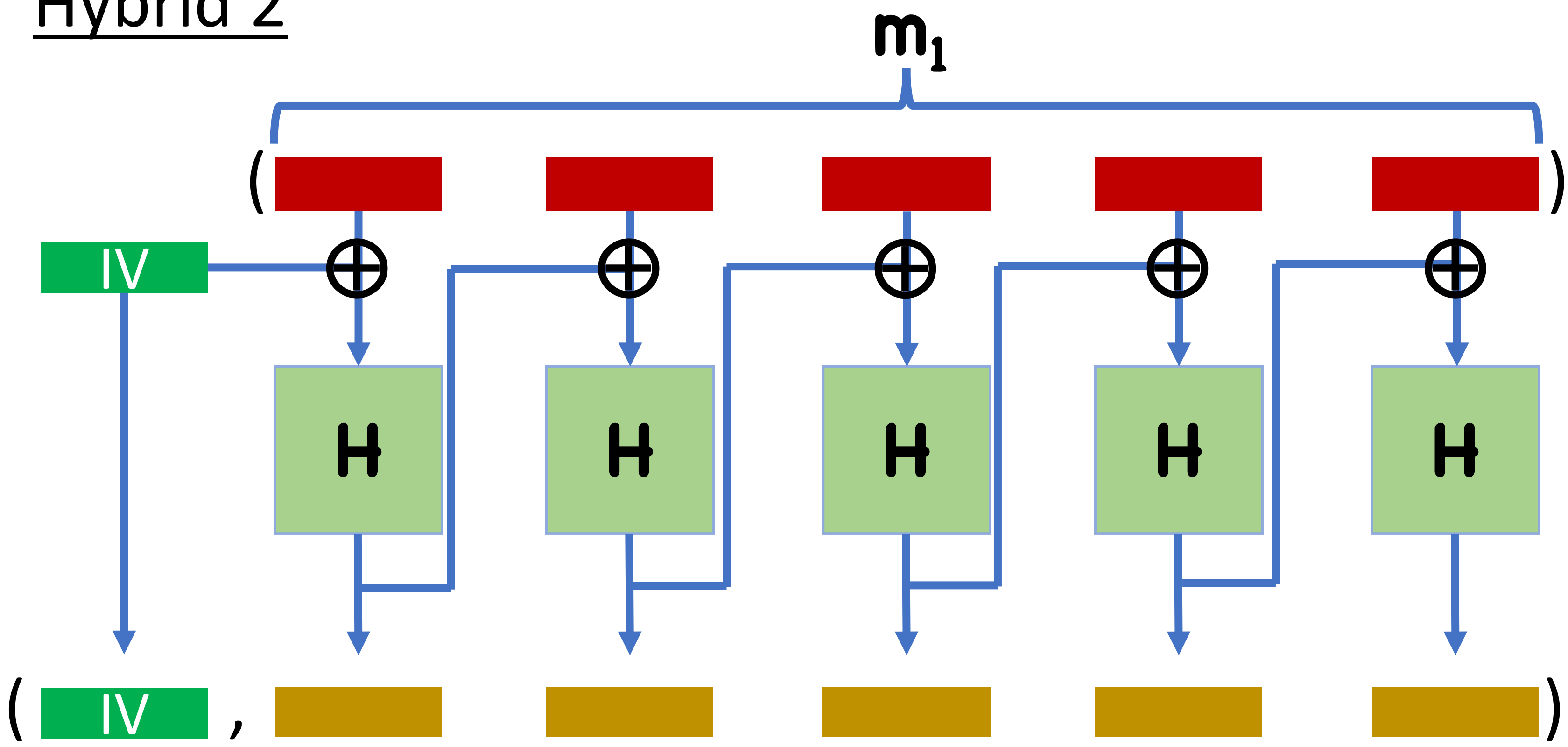
Proof Sketch

Hybrid 1



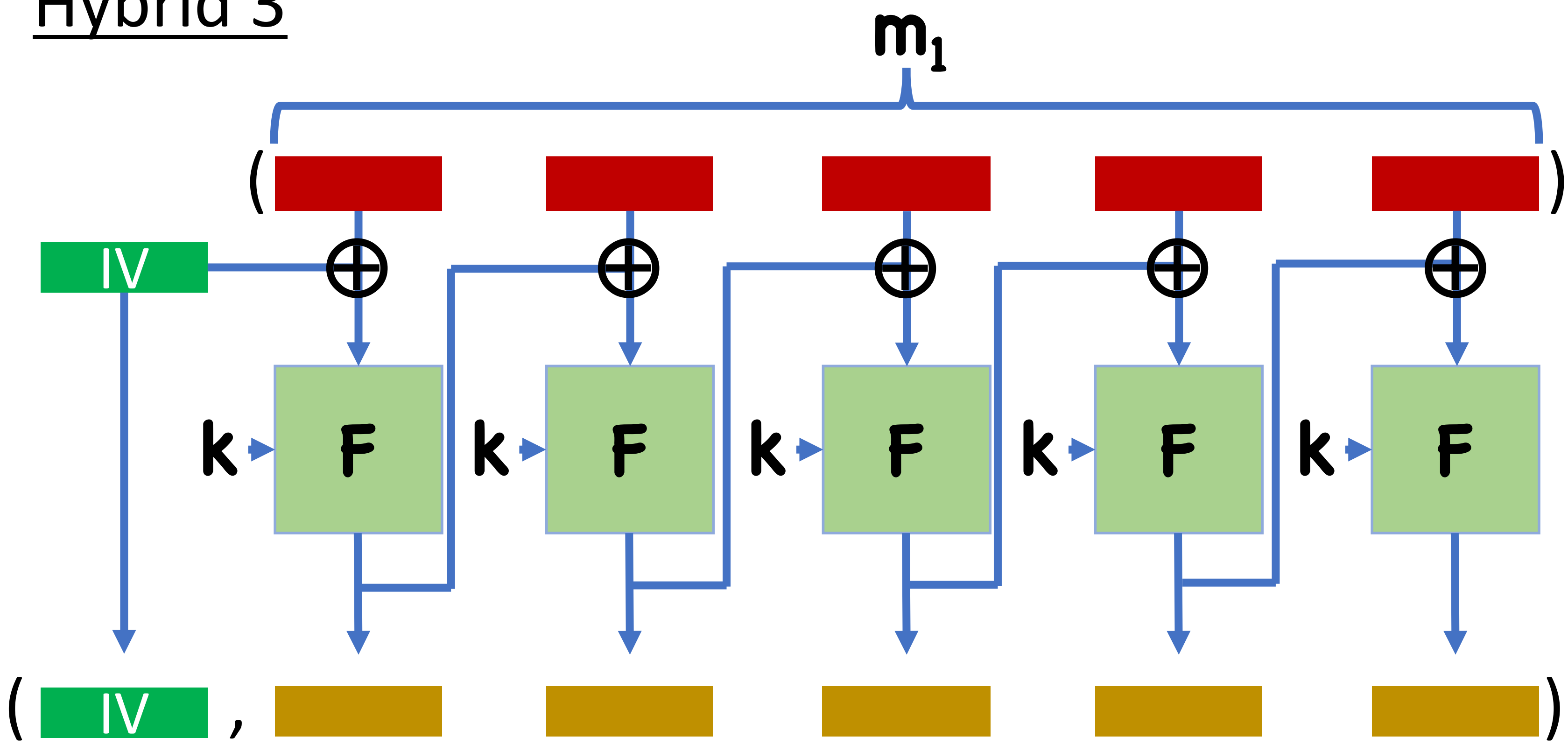
Proof Sketch

Hybrid 2



Proof Sketch

Hybrid 3



Proof Sketch

Hybrid 0,1 differ by replacing calls to **F** with calls to random permutation **H**

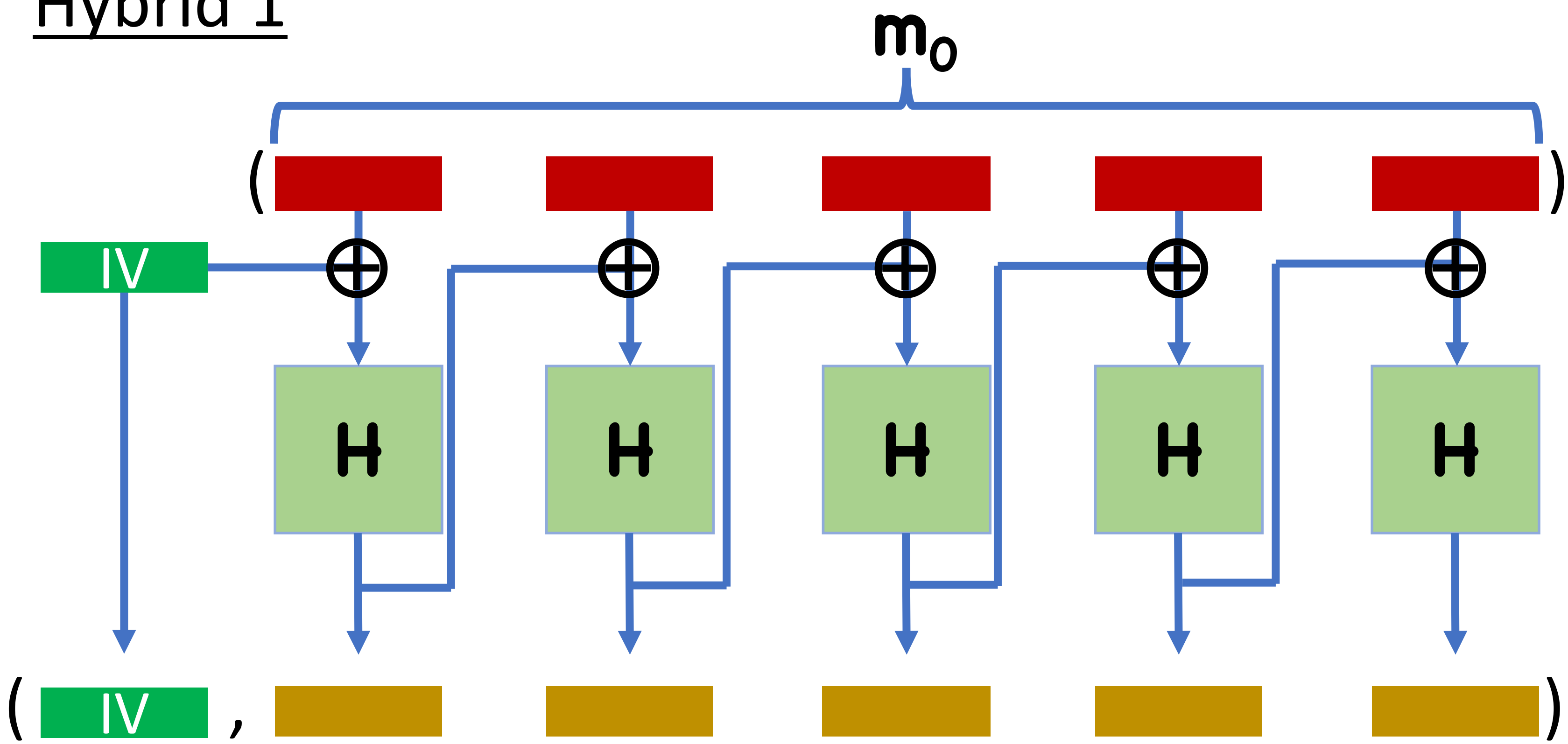
- Indistinguishable by PRP security

Same for Hybrids 2,3

All that is left is to show indistinguishability of 1,2

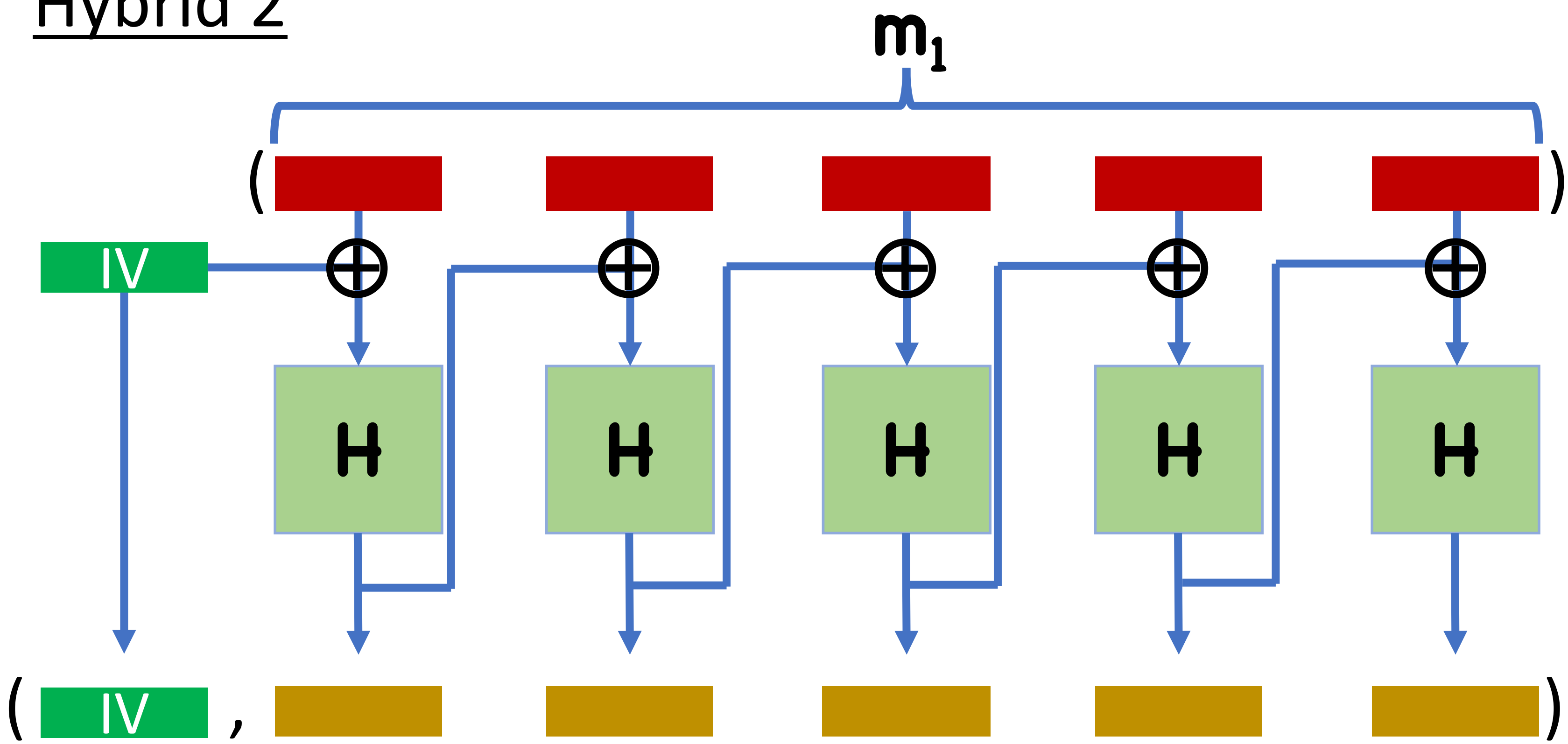
Proof Sketch

Hybrid 1



Proof Sketch

<u>Hybrid 2</u>



Proof Sketch

Idea:

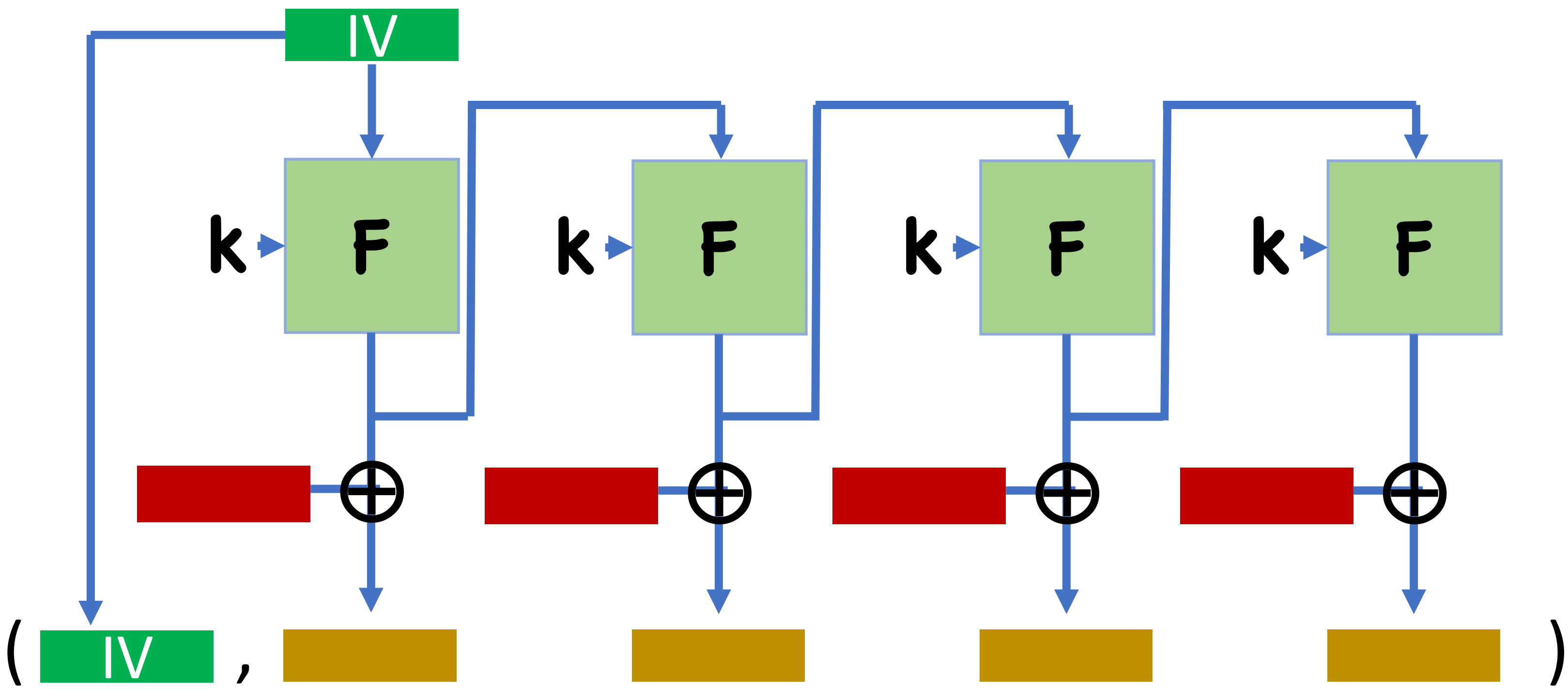
- As long as, say, the sequence of left messages queried by  does not result in two calls to **H** on the same input, all outputs will be random (distinct) outputs
- For each message, first query to **H** will be uniformly random
- Second query gets XORed with output of first query to **H** $\Rightarrow$ ≈ uniformly random

Proof Sketch

Idea:

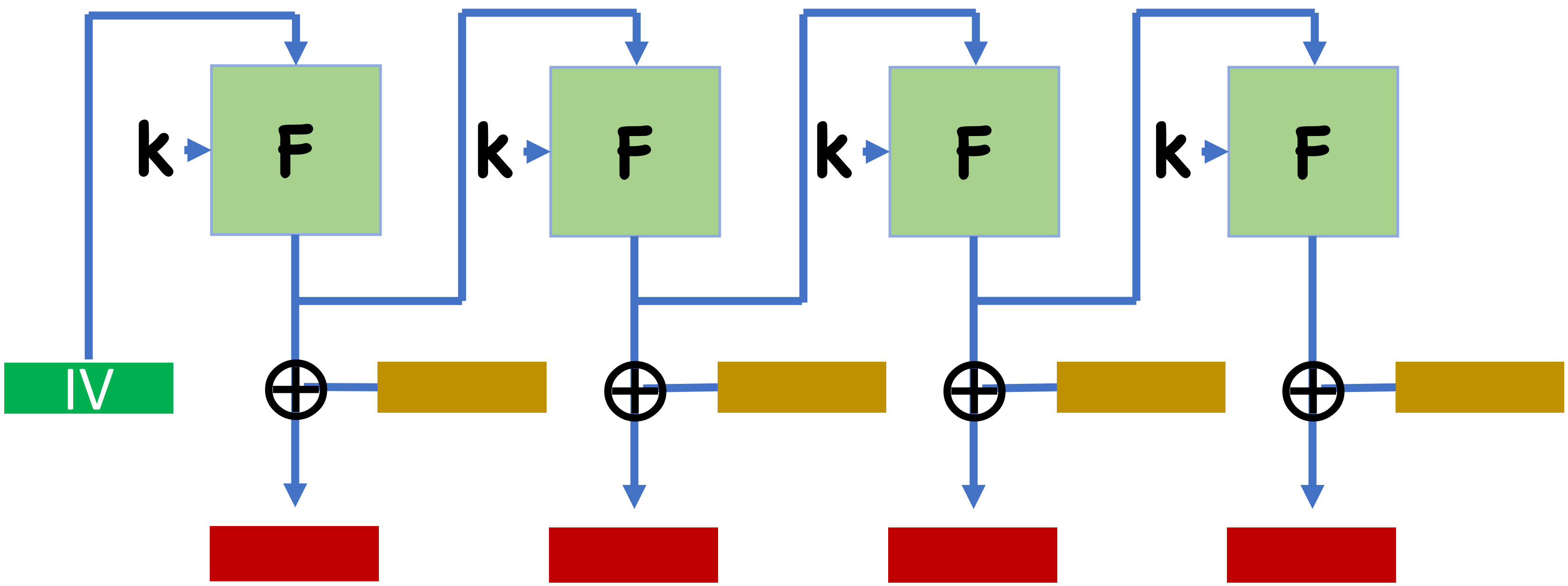
- Since queries to **H** are (essentially) uniformly random, probability of querying same input twice is exponentially small
- Ciphertexts will be essentially random
- True regardless of encrypting $\mathbf{m_0}$ or $\mathbf{m_1}$

Output Feedback Mode (OFB)

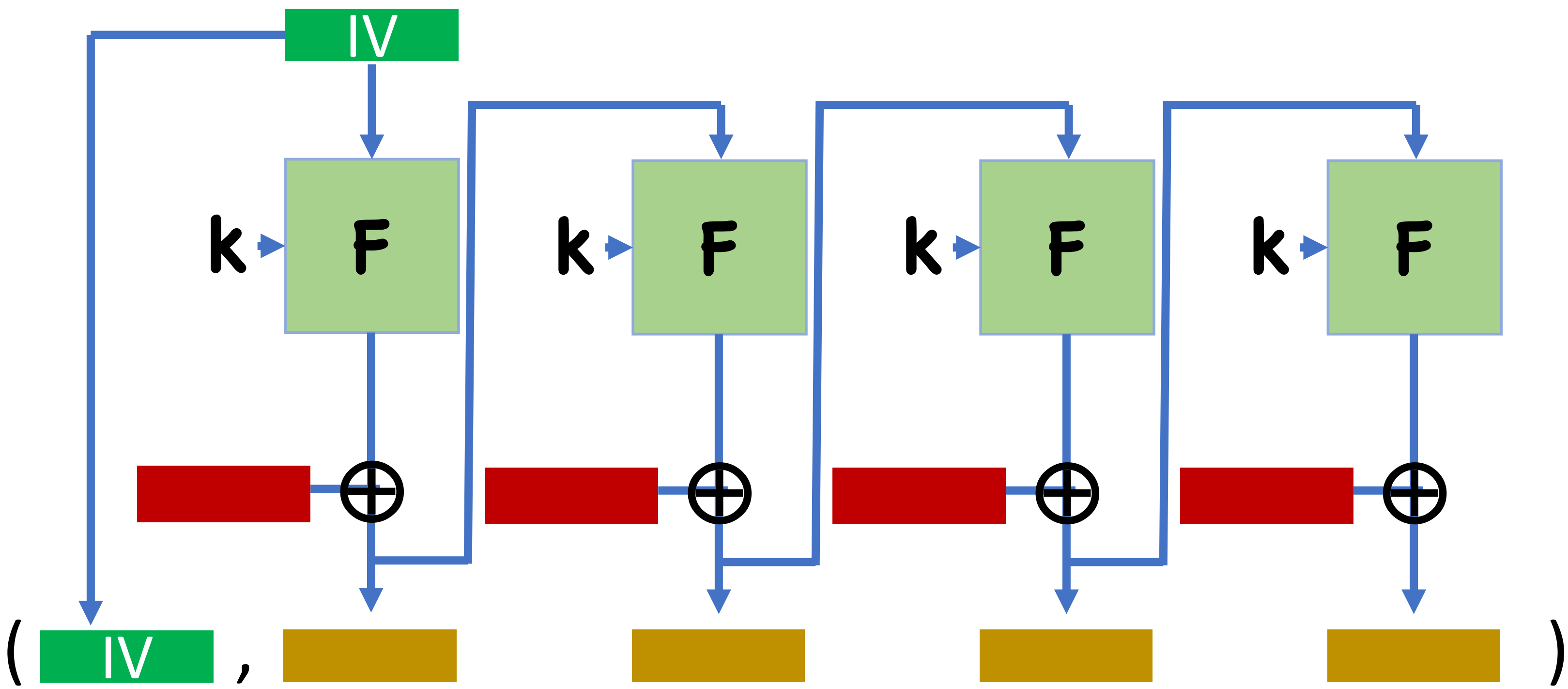


Turn block cipher into stream cipher

OFB Decryption

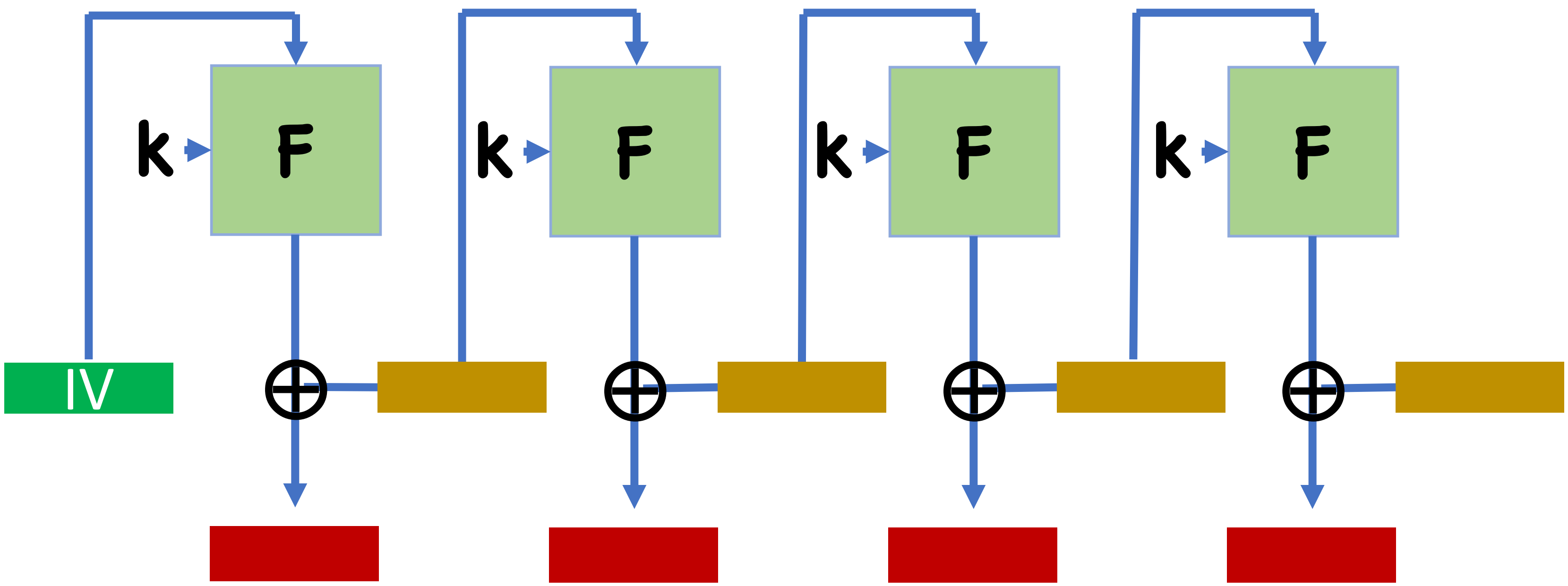


Cipher Feedback (CFB)



Turn block cipher into **self-synchronizing** stream cipher

CFB Decryption



Security of OFB, CFB modes

Security very similar to CBC

Define 4 hybrids
- 0: encrypt left messages
- 1: replace PRP with random permutation
- 2: encrypt right messages
- 3: replace random permutation with PRP

0,1 and 2,3 are indistinguishable by PRP security

1,2 are indistinguishable since ciphertexts are essentially random

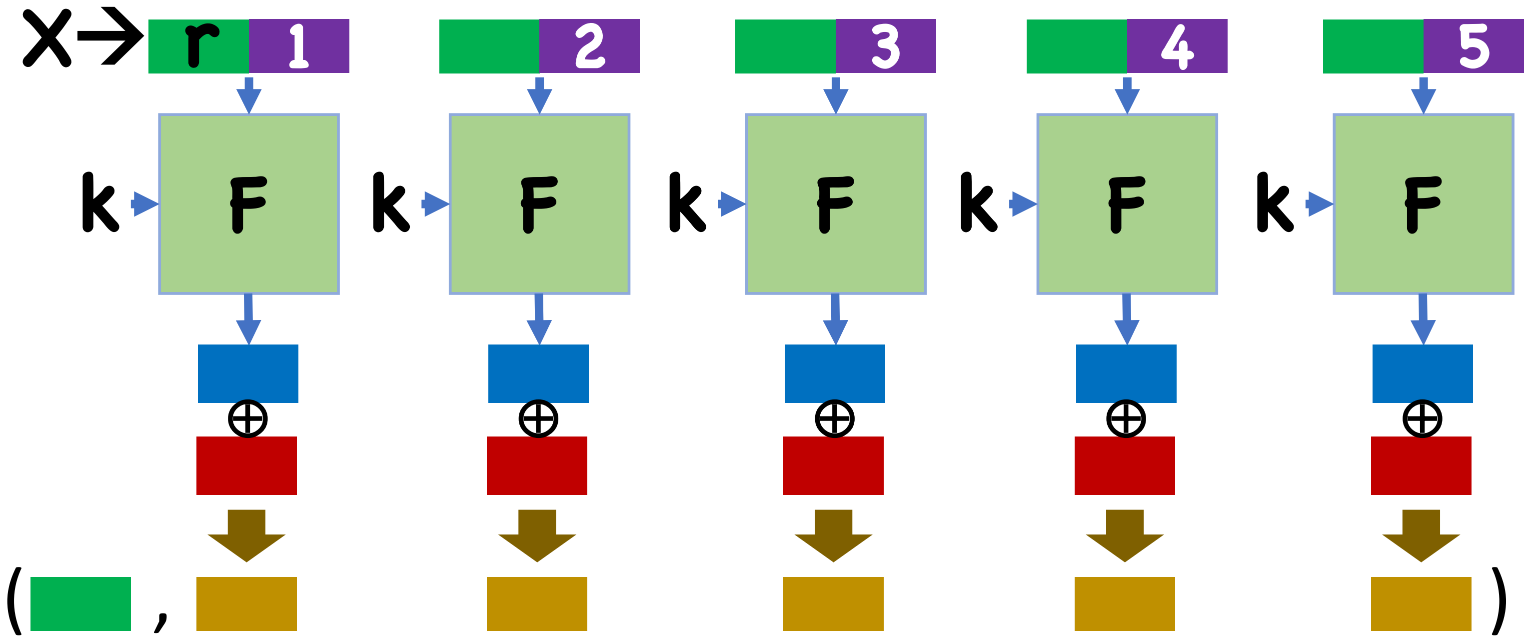
Which Mode to Use?

Never use ECB

Otherwise, largely depends on application
- Some advantages/disadvantages to each

Parallelism

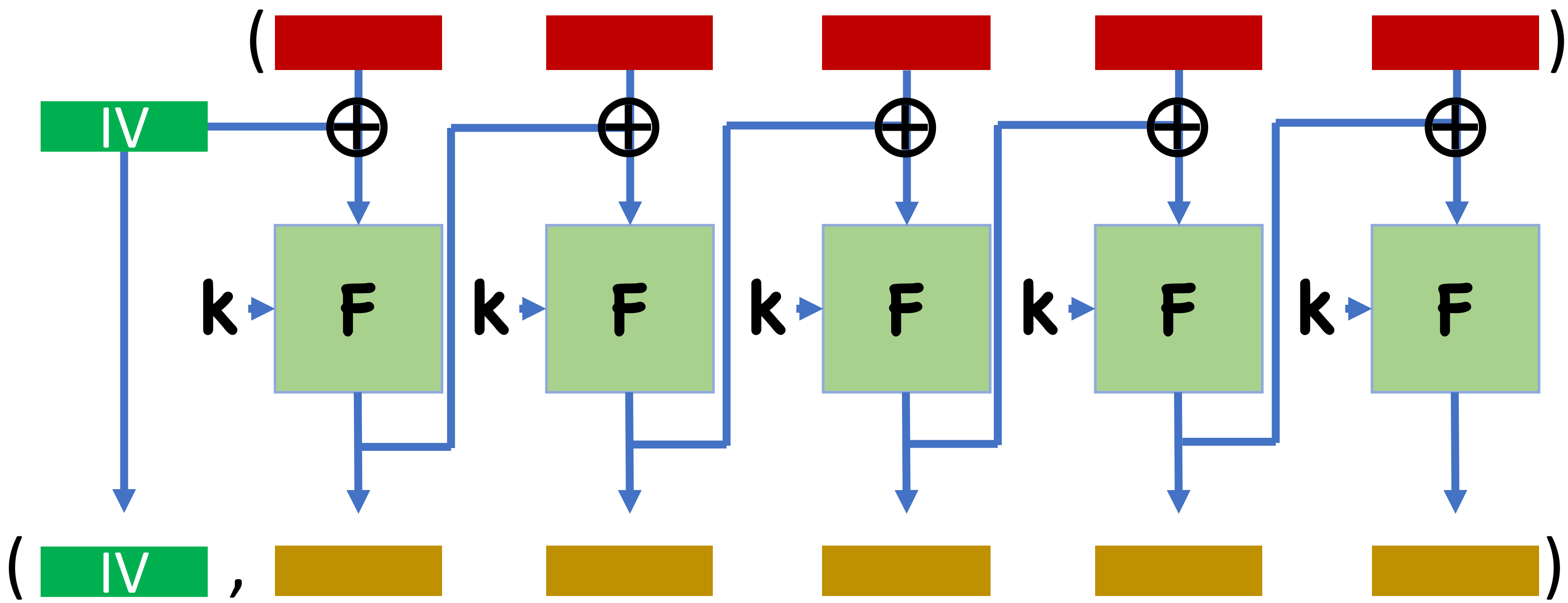
CTR mode:



Enc,Dec easily parallelized ✓

Parallelism

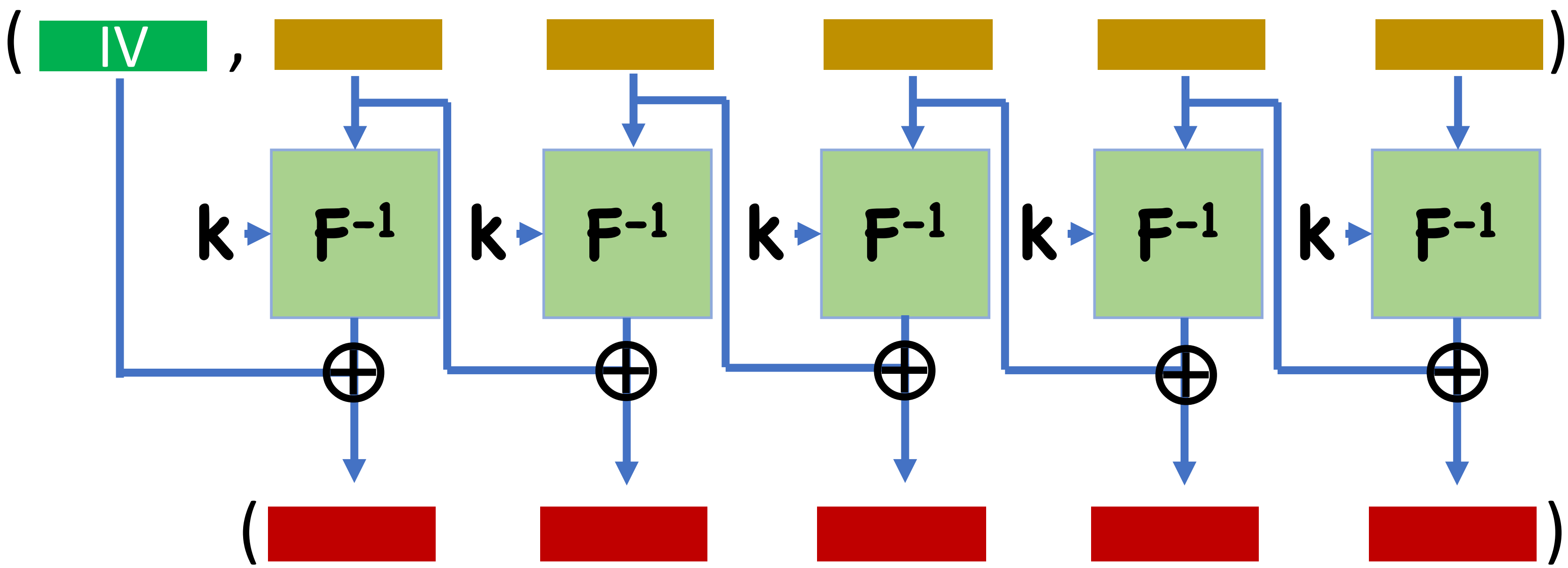
CBC mode encryption:



Enc not parallelizable ✗

Parallelism

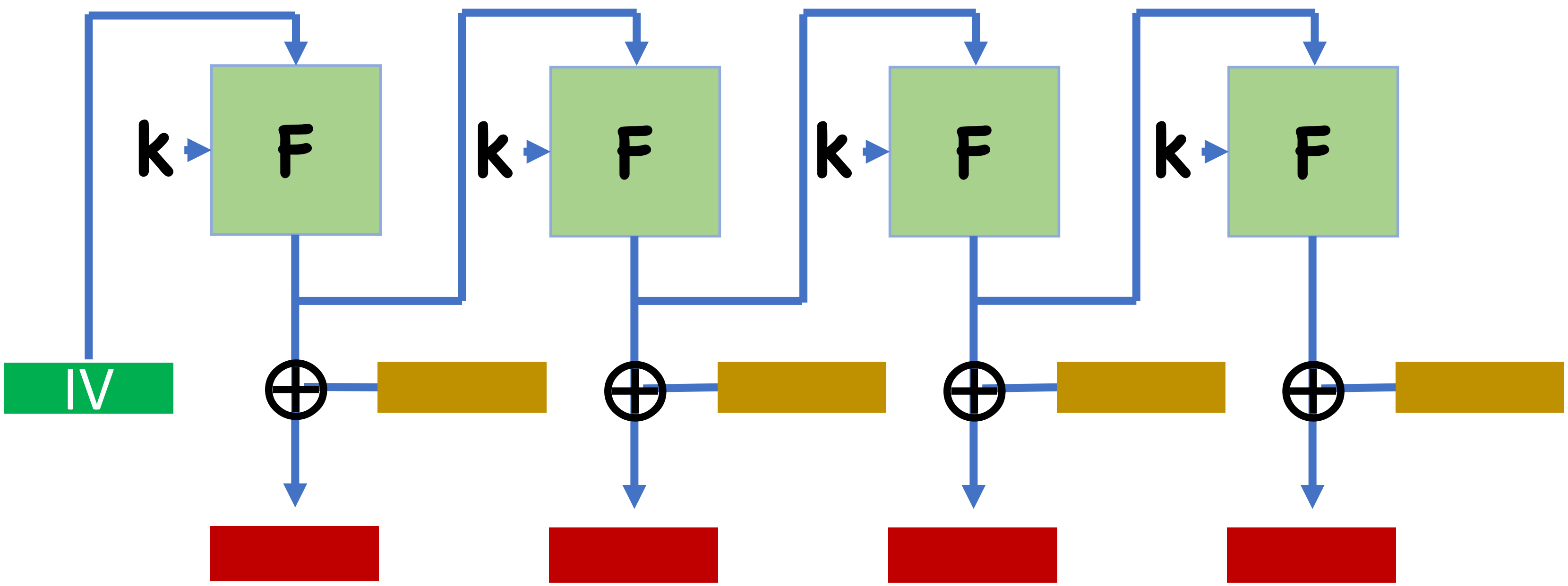
CBC mode decryption:



Dec parallelizable ✓

Parallelism

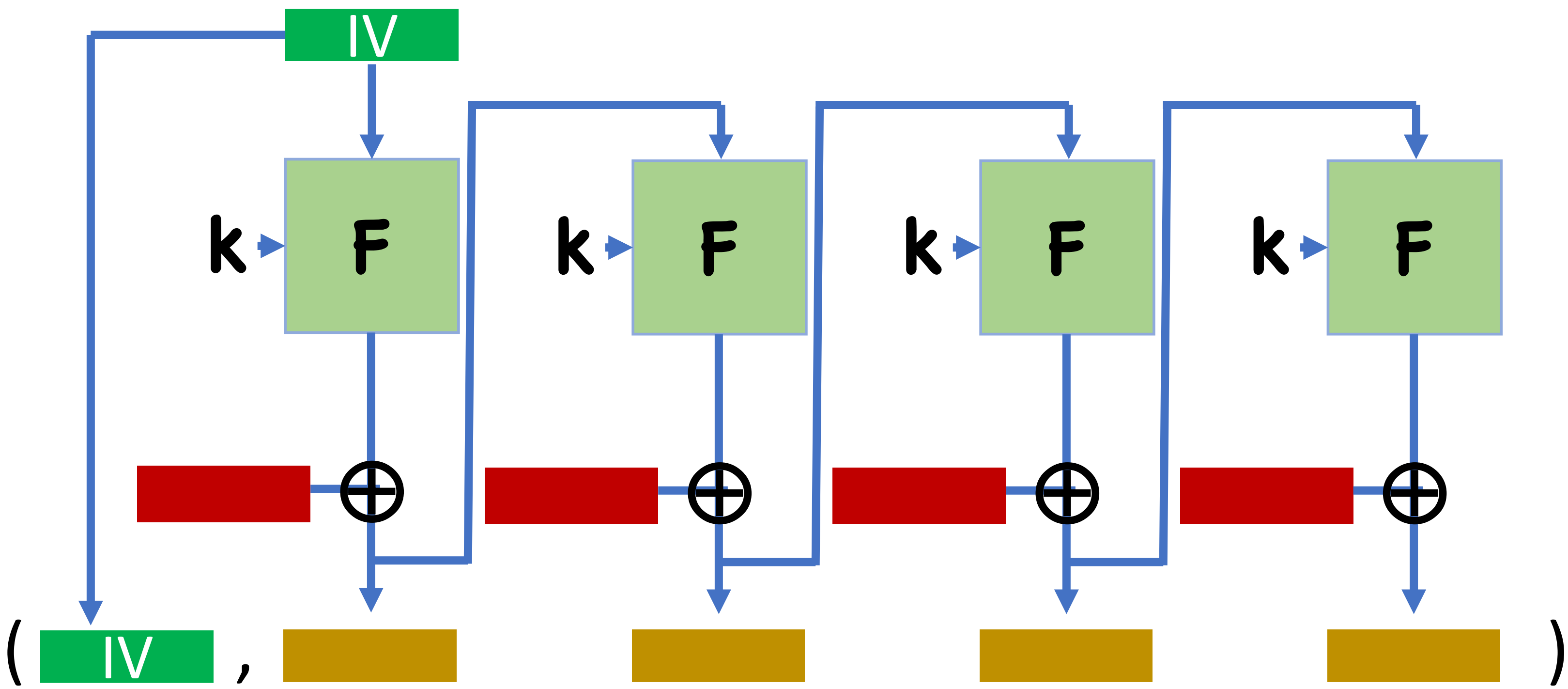
OFB mode:



Enc,Dec not parallelizable ✗

Parallelism

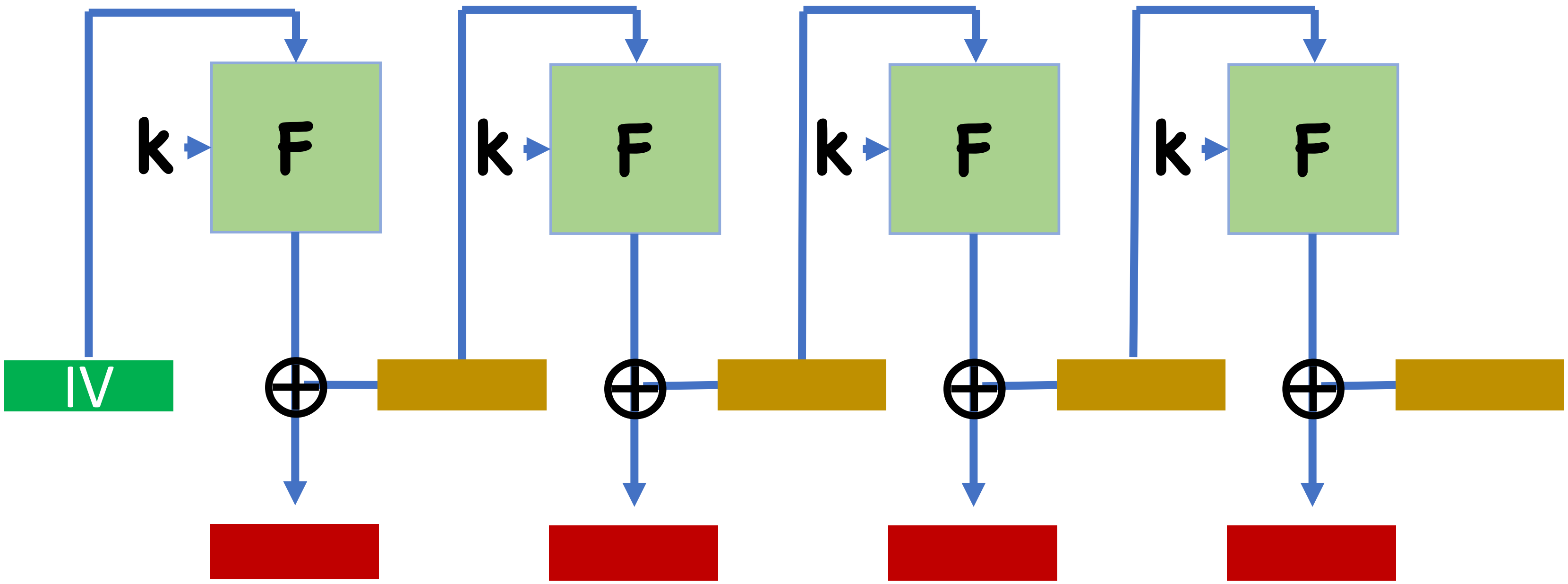
CFB mode encryption:



Enc not parallelizable ✗

Parallelism

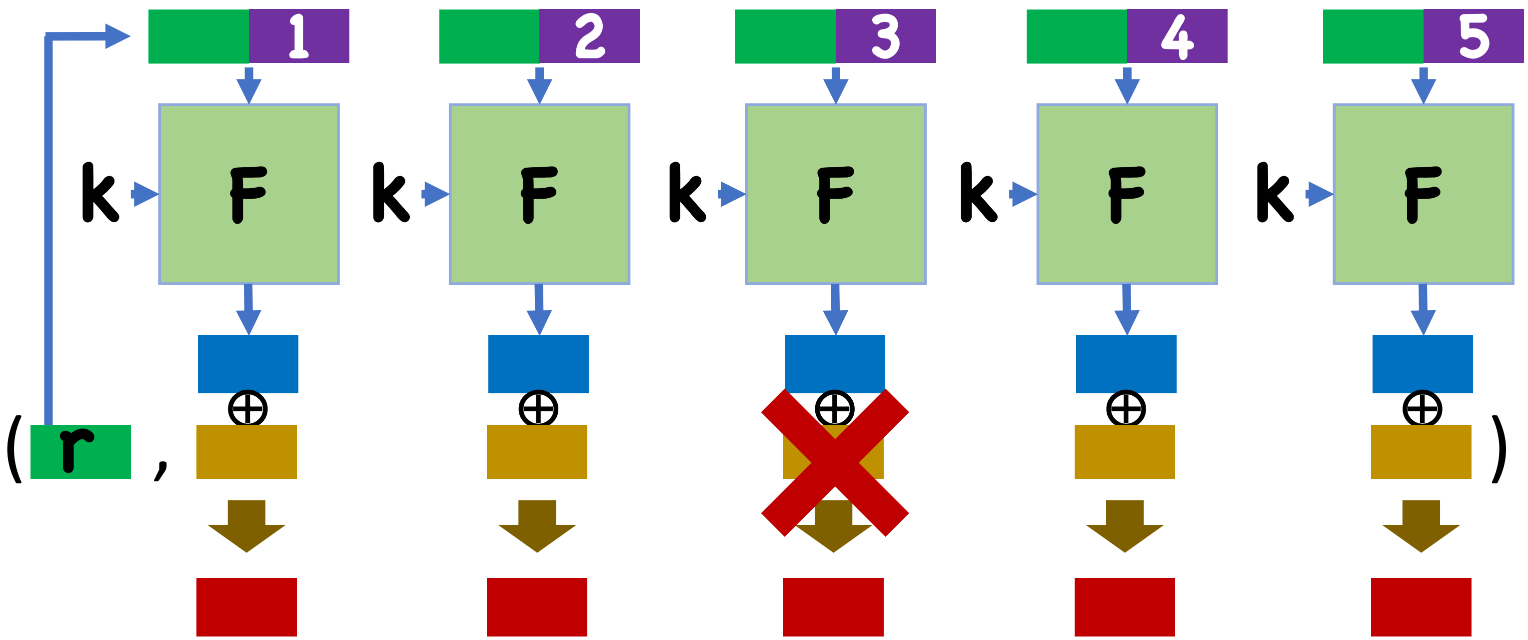
CFB mode decryption:



Dec parallelizable ✓

Lose Block During Transmission?

CTR mode decryption:

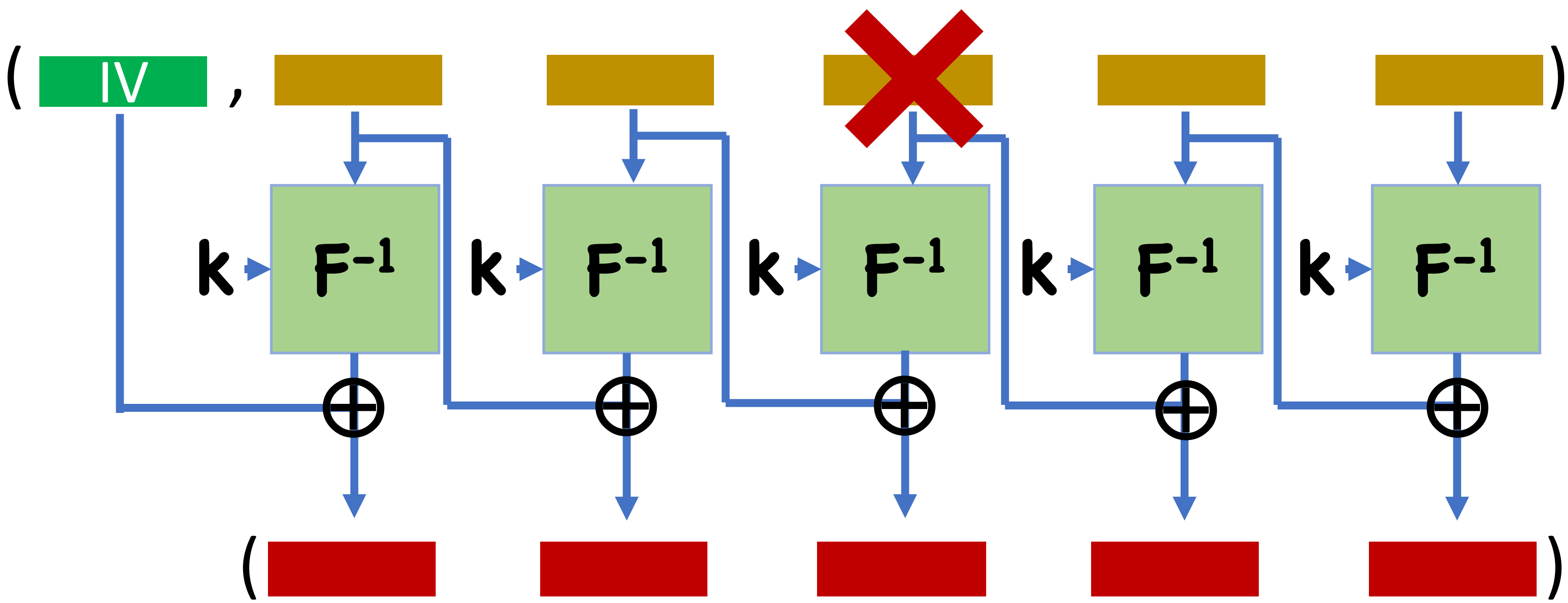


Message corrupted after deleted block 

Same for any mode that builds stream cipher (e.g. OFB)

Lose Block During Transmission?

CBC mode decryption:



Lose one block, one more corrupted, rest fine ✓

Same for CFB

PRPs vs PRFs

In practice, PRPs are the central building block of most crypto

- Also PRFs
- Can build PRGs
- Very versatile

Constructing block ciphers

Difficulties

$2^n!$ Permutations on n-bit blocks

$\Rightarrow \approx n2^n$ bits to write down random perm.

Reasonable for very small n (e.g. $n<20$), but totally infeasible for large n (e.g. $n=128$)

Challenge:

- Design permutations with small description that "behave like" random permutations

Difficulties

For a random permutation **H**, **H(x)** and **H(x')** are (essentially) independent random strings

- Even if **x** and **x'** differ by just a single bit

Therefore, for a random key **k**, changing a single bit of **x** should "affect" all output bits of **F(k,x)**

Definition: For a function $\mathbf{H:\{0,1\}^n \to \{0,1\}^n}$, we say that bit $\mathbf{i}$ of the input affects bit $\mathbf{j}$ of the output if

For a random $\mathbf{x_1,\dots,x_{i-1},x_{i+1}, \dots, x_n}$, if we let
$\mathbf{y=H(x_1\dots x_{i-1}0x_{i+1}\dots x_n)}$ and
$\mathbf{z=H(x_1\dots x_{i-1}1x_{i+1}\dots x_n)}$
Then $\mathbf{y_j \neq z_j}$ with probability $\mathbf{\approx 1/2}$

Theorem: If (F, F^{-1}) is a secure PRP, then with (with "high" probability over the key k), for the function $F(k, \bullet)$, every bit of input affects every bit of output

Proof sketch:

- For random permutations this is true
- If bit i did not affect bit j, we can construct an adversary that distinguishes F from random

Confusion/Diffusion Paradigm

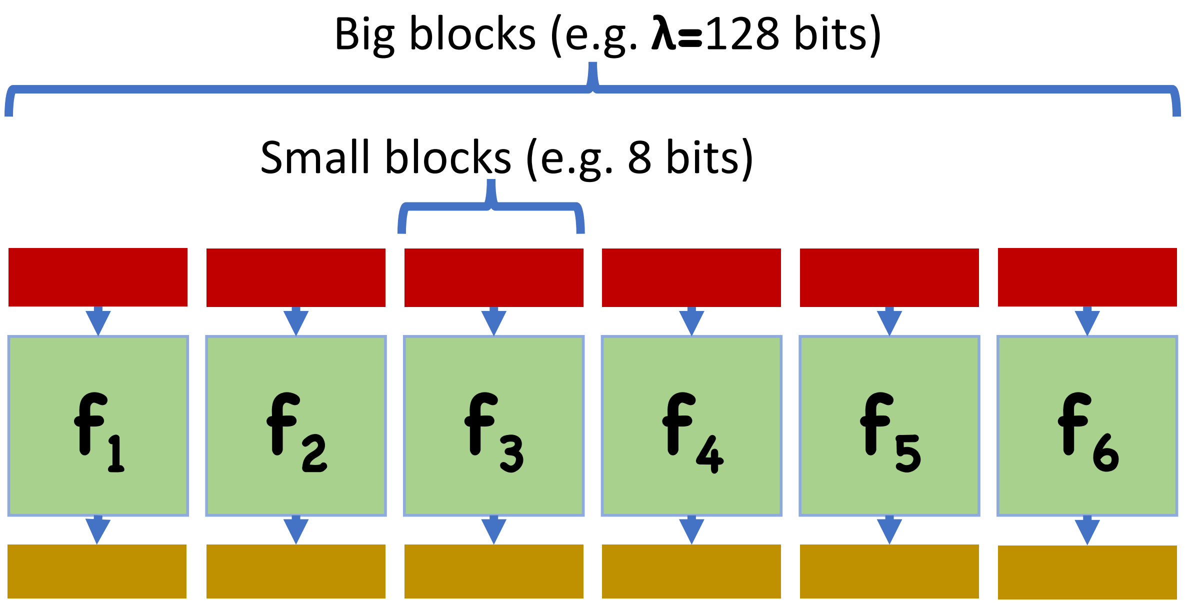
Confusion/Diffusion Paradigm

Goal: build permutation for large blocks from permutations for small blocks

- Small block perms can be made truly random
- Hopefully result is pseudorandom

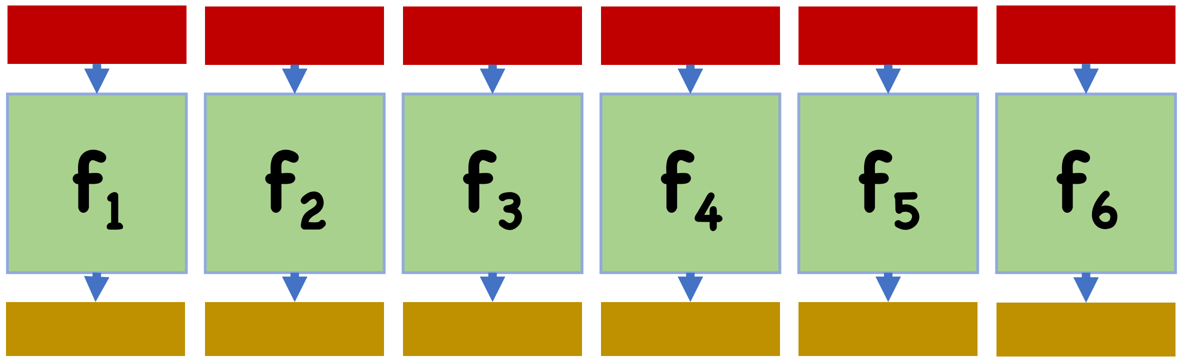
Confusion/Diffusion Paradigm

First attempt: break blocks into smaller blocks, apply smaller permutation blockwise



Key: description of f_1, f_2,…

Confusion/Diffusion Paradigm

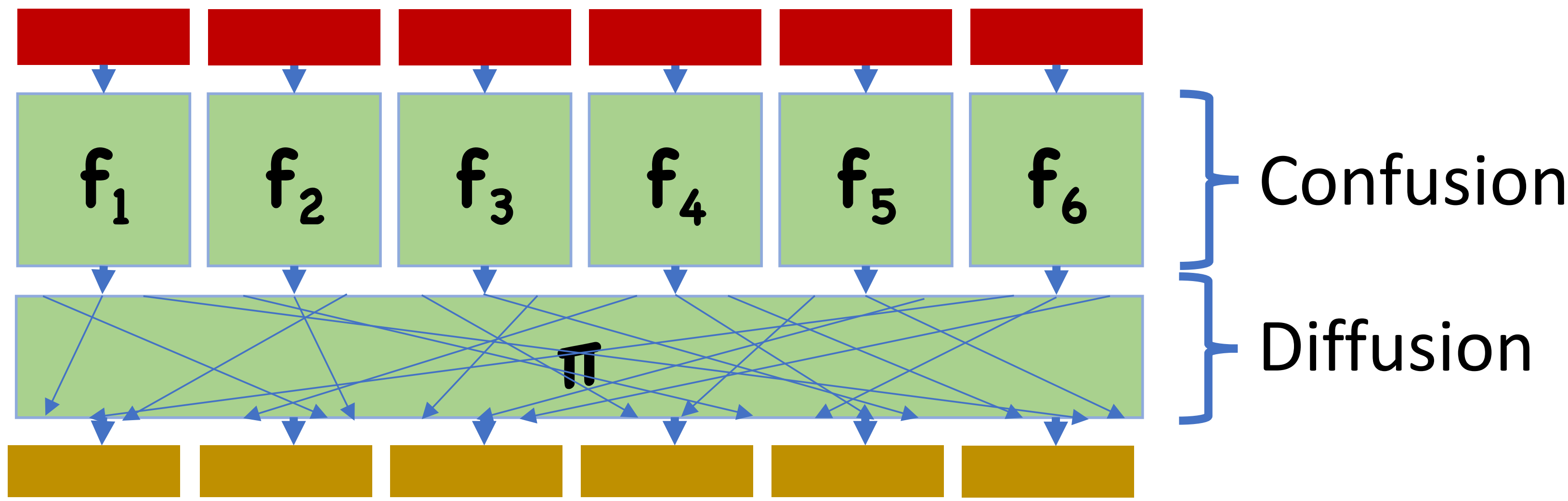


Is this a secure PRP?

- Key size: $\approx(8\times2^8)\times(\lambda/8) = O(\lambda)$
- Running time: a few table lookups, so efficient
- Security?

Confusion/Diffusion Paradigm

Second attempt: shuffle output bits

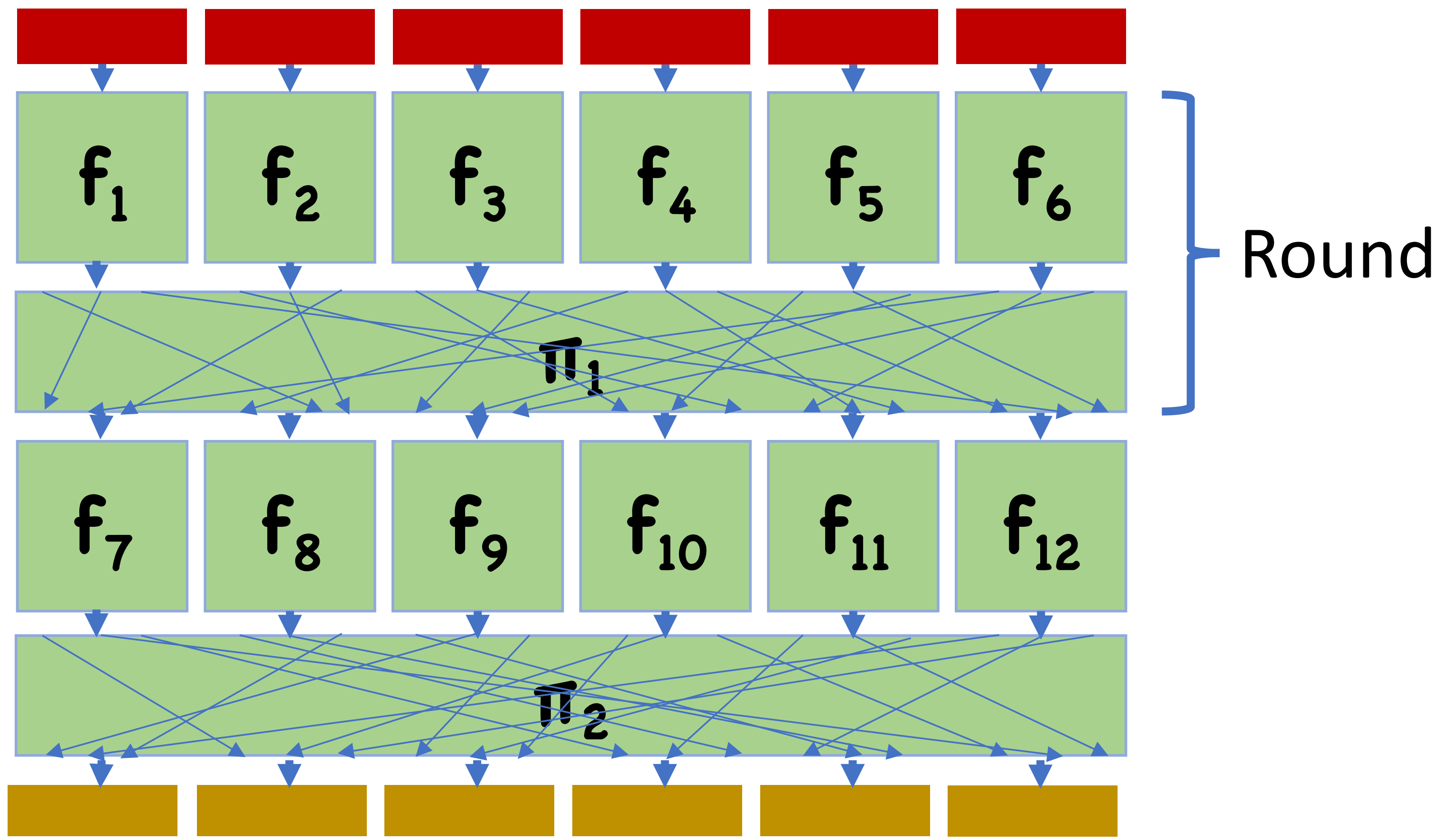


Is this a secure PRP?

- Key size: **$\approx 2^8\lambda + \lambda\times\log\lambda$**
- Running time: a few table lookups
- Security?

Confusion/Diffusion Paradigm

Third Attempt: Repeat multiple times!



Confusion/Diffusion Paradigm

While single round is insecure, we've made progress
- Each bit affects 8 output bits

With repetition, hopefully we will make more and more progress

Confusion/Diffusion Paradigm

With 2 rounds,

- Each bit affects 64 output bits

With 3 rounds, all 128 bits are affected

Repeat a few more times for good measure

Announcements/Reminders

HW2 due September 29

- Submit through Gradescope

PR1 Due October 6