COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Fall 2020

Announcements/Reminders

Last day to submit HW2

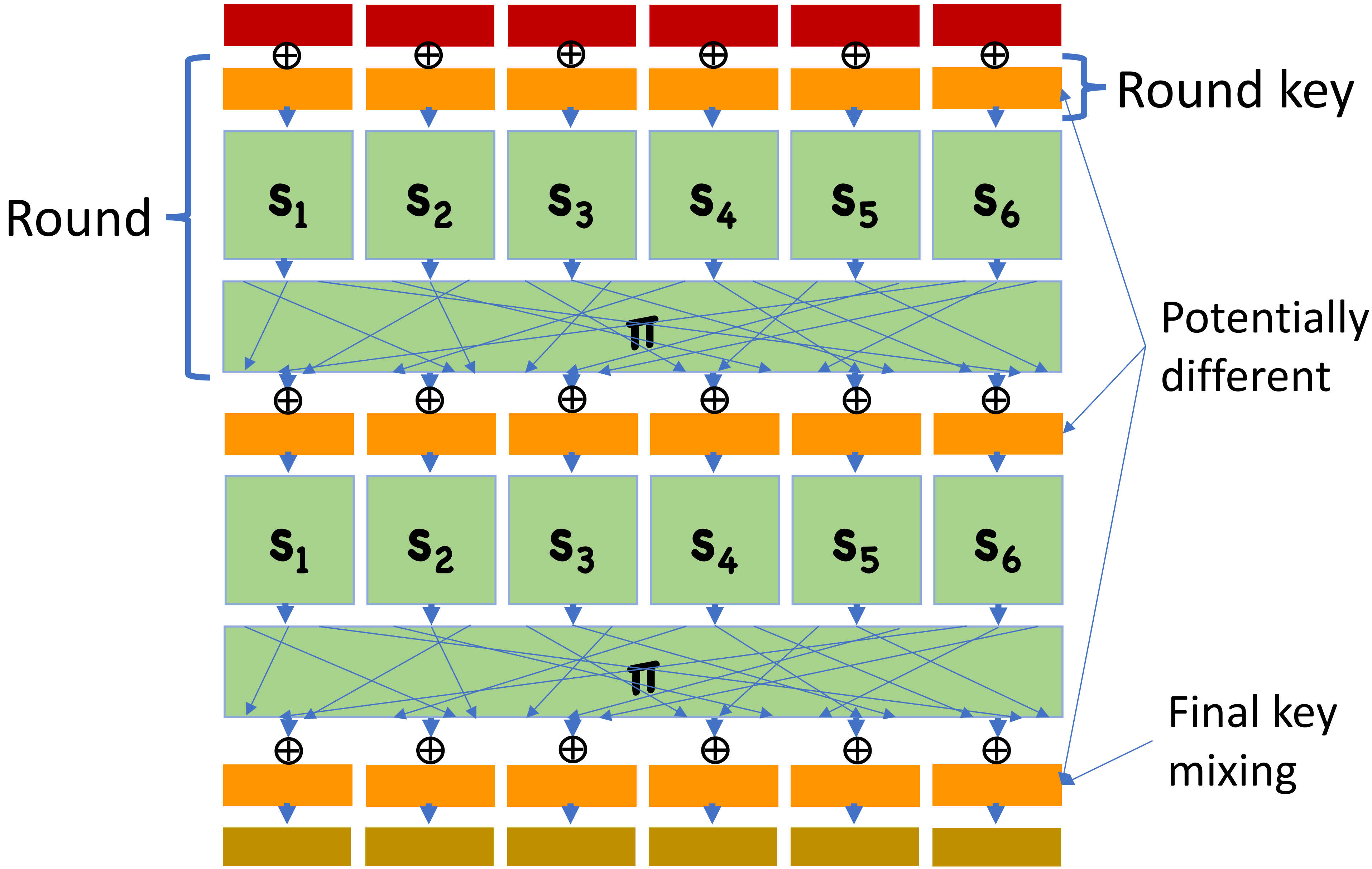
- Submit through Gradescope

PR1 Due October 6

Heads up: HW3 will be out soon, due on Oct 20

Previously on COS 433…

Substitution Permutation Networks



Substitution Permutation Networks

To specify a network, must:

- Specify S-boxes
- Specify P-box
- Specify key schedule (how round keys are derived from master)

Choice of parameters can greatly affect security

Attacks on block ciphers

Meet In The Middle Attacks

k_0	$d = DES(k_0,m)$
0	52
1	93
2	03
3	96
4	20
5	49
...	...

k_1	$d = DES^{-1}(k_1,m)$
0	69
1	10
2	86
3	49
4	99
5	08
...	...

Meet In The Middle Attacks

Complexity of meet in the middle attack:

- Computing two tables: time, space $2 \times 2^{\text{key length}}$
- Slight optimization: don't need to actually store second table

On 2DES, roughly same time complexity as brute force on DES

Generalizing MITM

In general, given **r** rounds of a block cipher with **t**-bit keys,

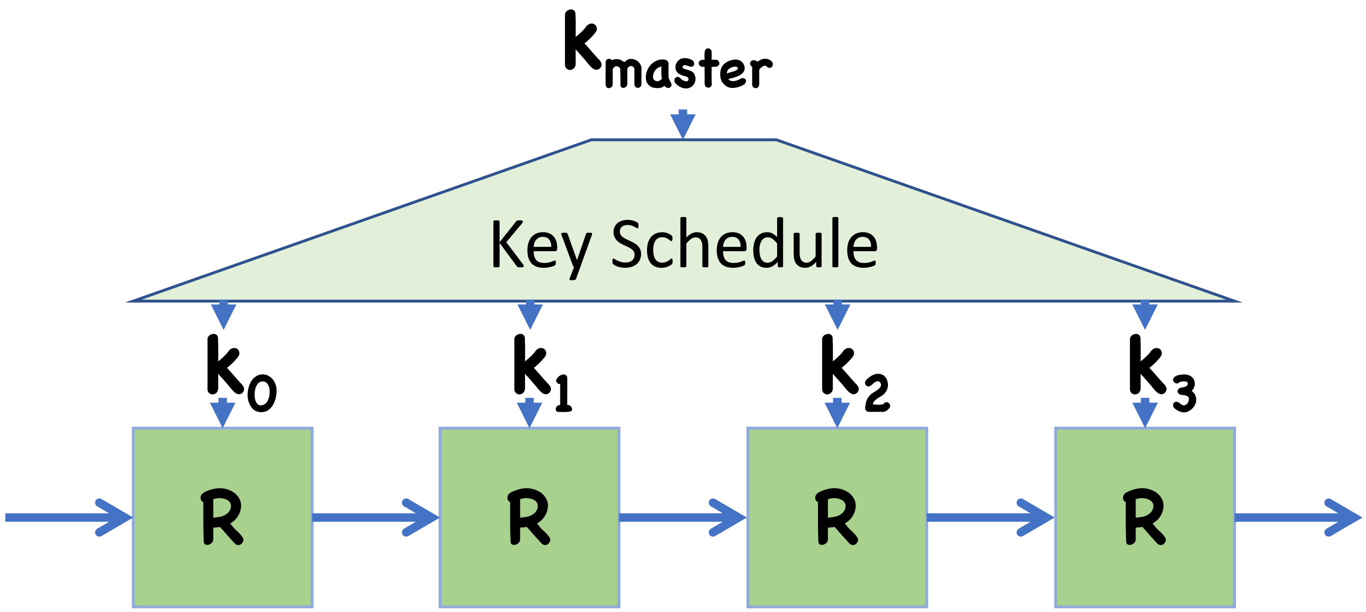
- Attack time: $2^{t\lceil r/2 \rceil}$
- Attack space: $2^{t\lfloor r/2 \rfloor}$

Today:

- More Attacks on block ciphers
- Message Integrity

MITM Attacks

MITM attacks can also be applied to plain single block ciphers



Can yield reasonable attacks in some regimes

Time-Space Tradeoffs

MITM attack requires significant space

In contrast, brute force requires essentially no space, but runs slower

Known as a time-space tradeoff

Another Time-Space Trade-off Example

Given $y=F(k,x)$, find x

- Allowed many queries to $F(k,x)$ oracle
  (That is, standard block cipher oracle)
- Assume $|k| \gg |x|$

Option 1:

- Brute force search over entire domain looking for x
- Time: 2^n $(n=|x|)$
- Space: 1

Another Time-Space Trade-off Example

Given $y=F(k,x)$, find x

- Allowed many queries to $F(k,x)$ oracle
  (That is, standard block cipher oracle)
- Assume $|k| >> |x|$

Option 2: Preprocessing

- Before seeing y, compute giant table of $(x,F(k,x))$ pairs, sorted by $F(k,x)$
- Preprocessing Time: 2^n
- Space: 2^n
- Online time: essentially constant

Option 3: Hellman's Attack

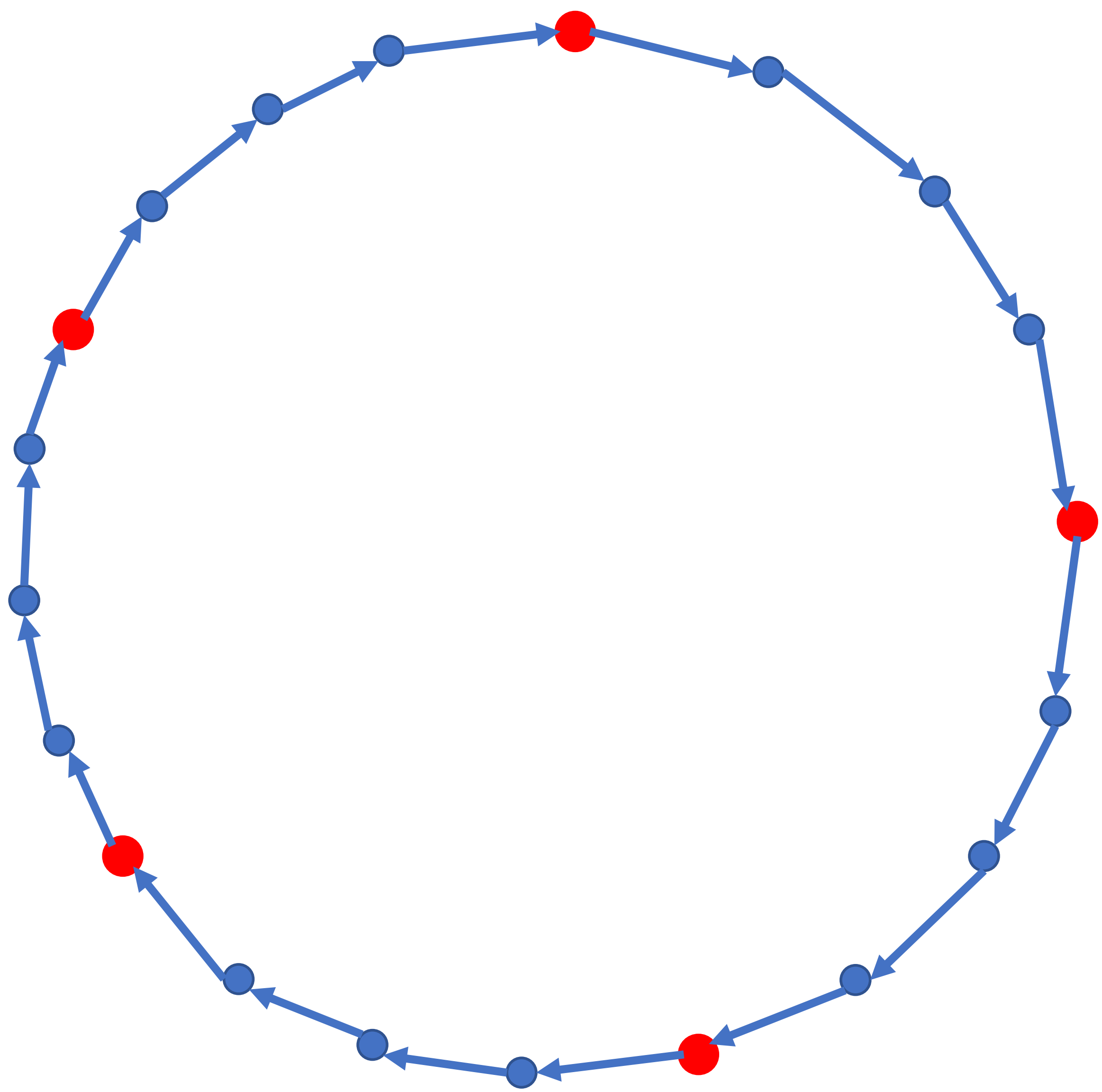
For simplicity, assume $F(k,\bullet)$ forms a cycle covering entire domain

- $\{0, F(k,0), F(k, F(k,0)), F(k, F(k, F(k,0))),\ldots\} = X$

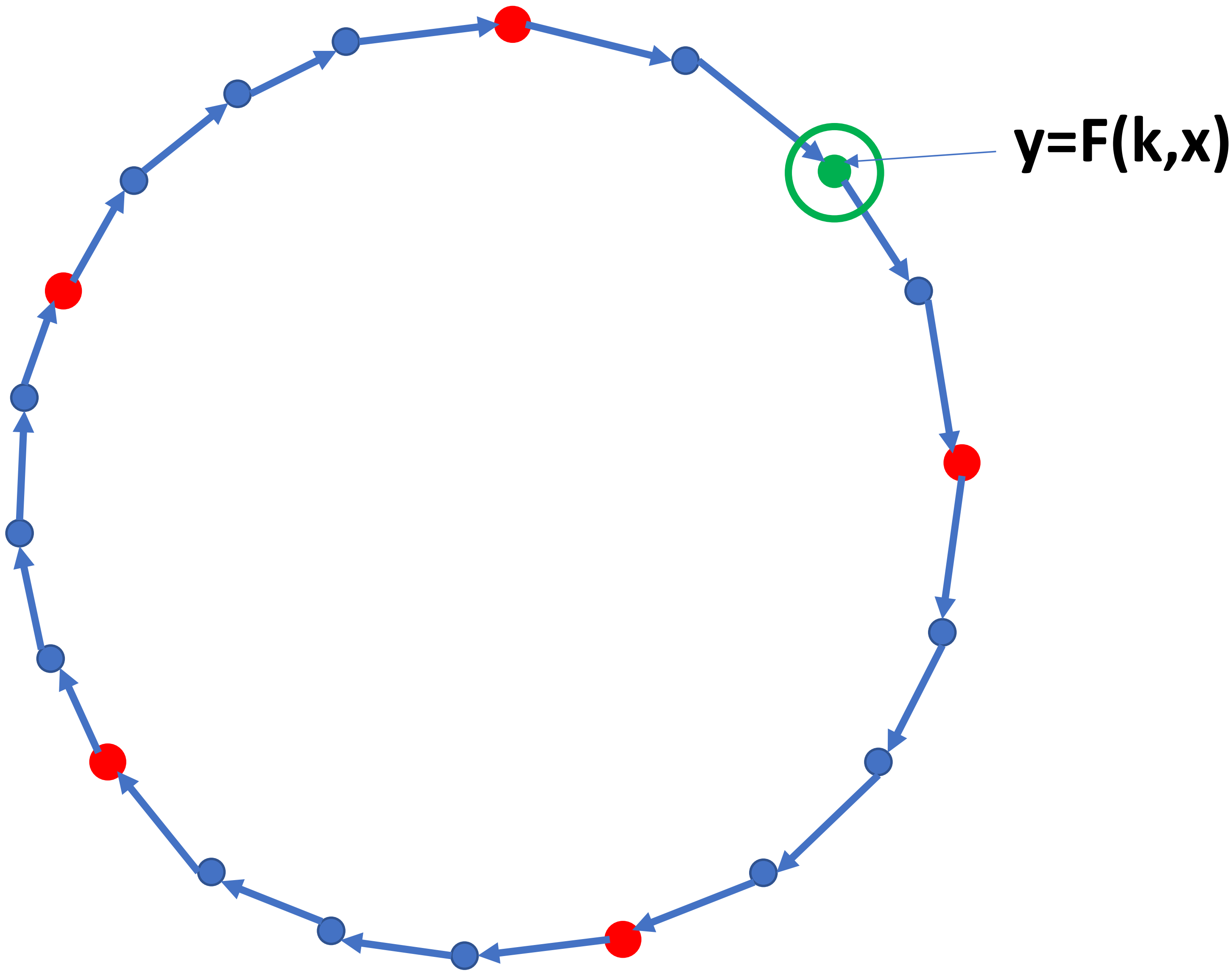
In preprocessing stage:

- Attacker iterates over entire cycle, saving every t^{th} term in a table $(x_1,\ldots,x_{N/t})$ where $N=2^n$

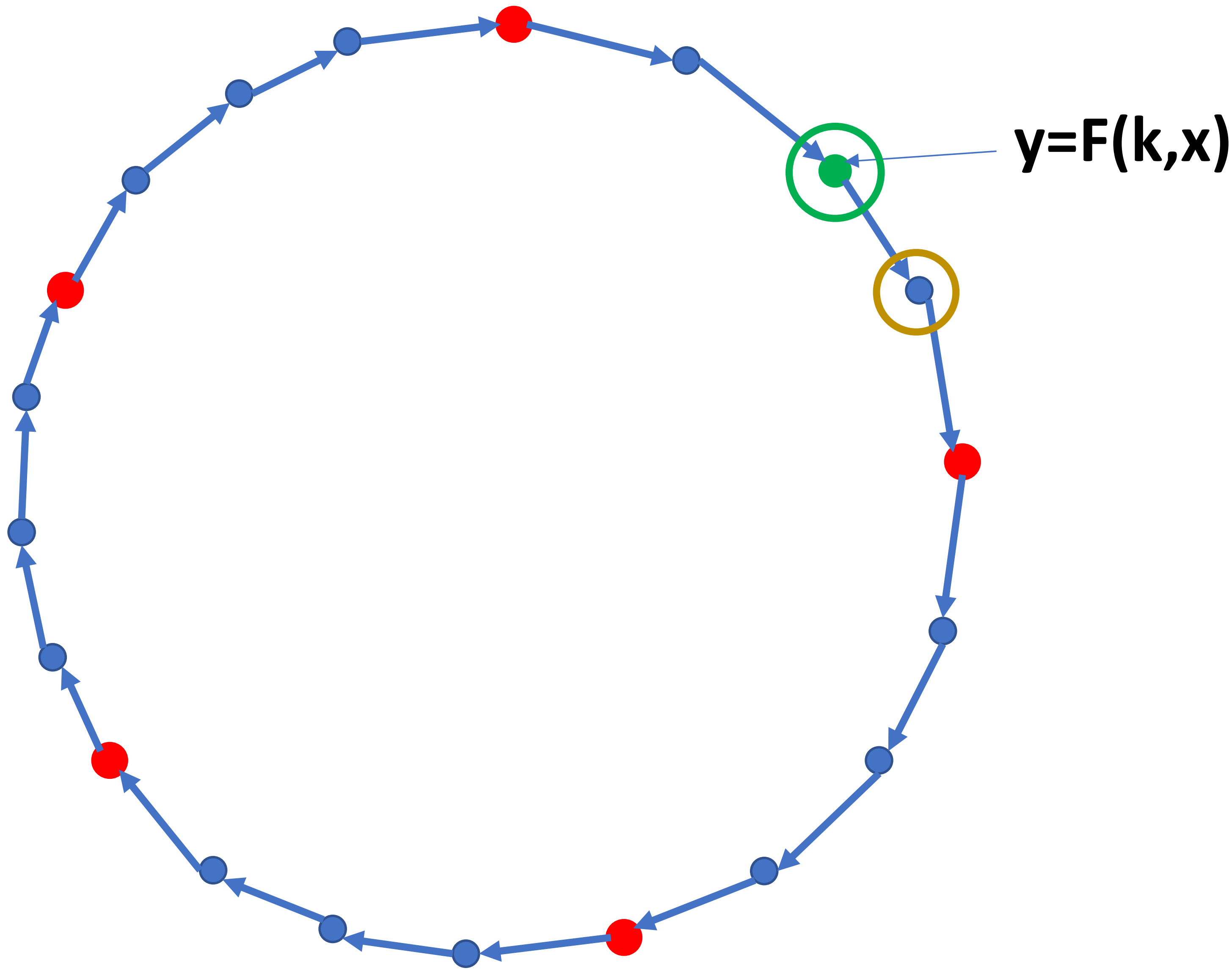
Option 3: Hellman's Attack



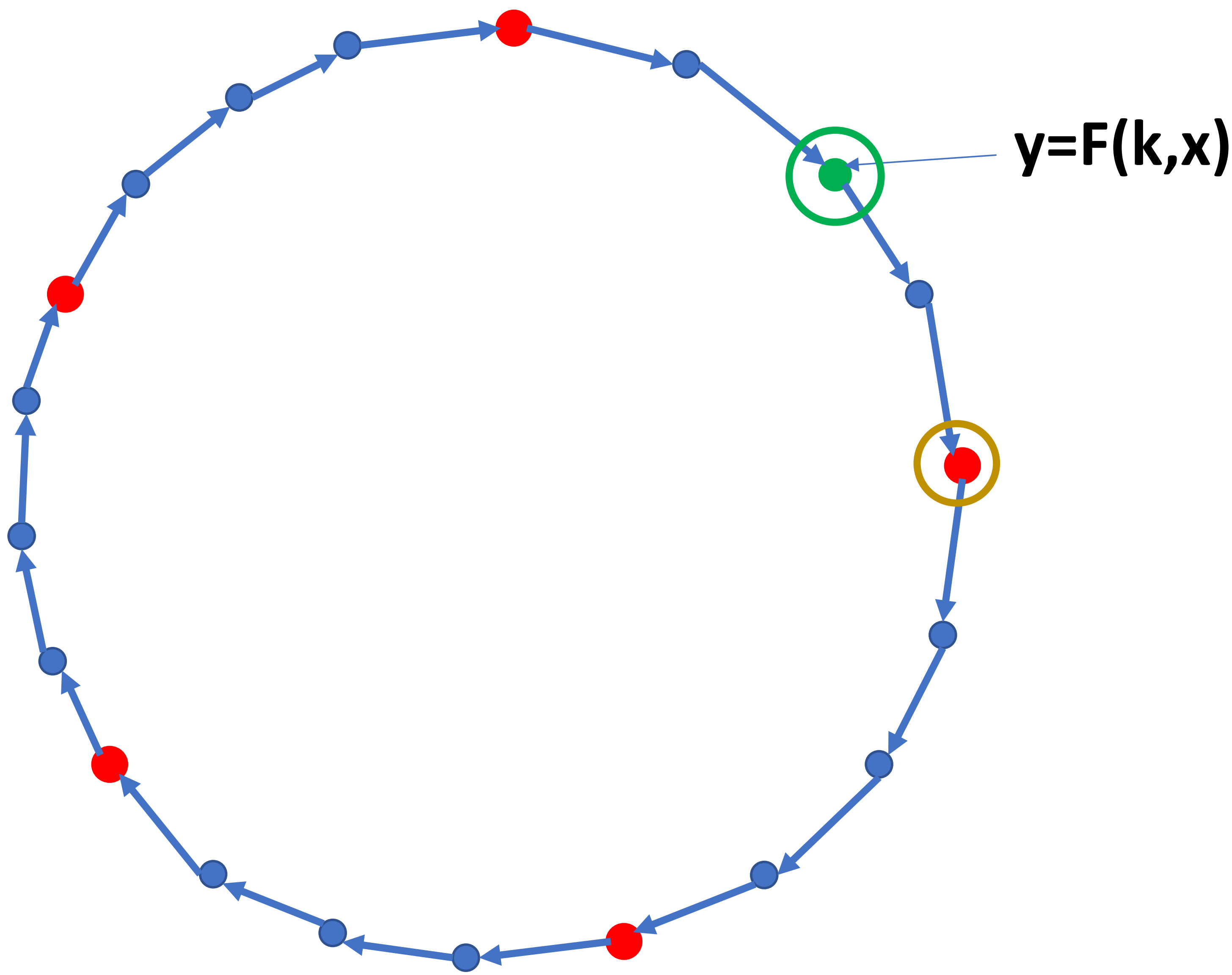
Option 3: Hellman's Attack



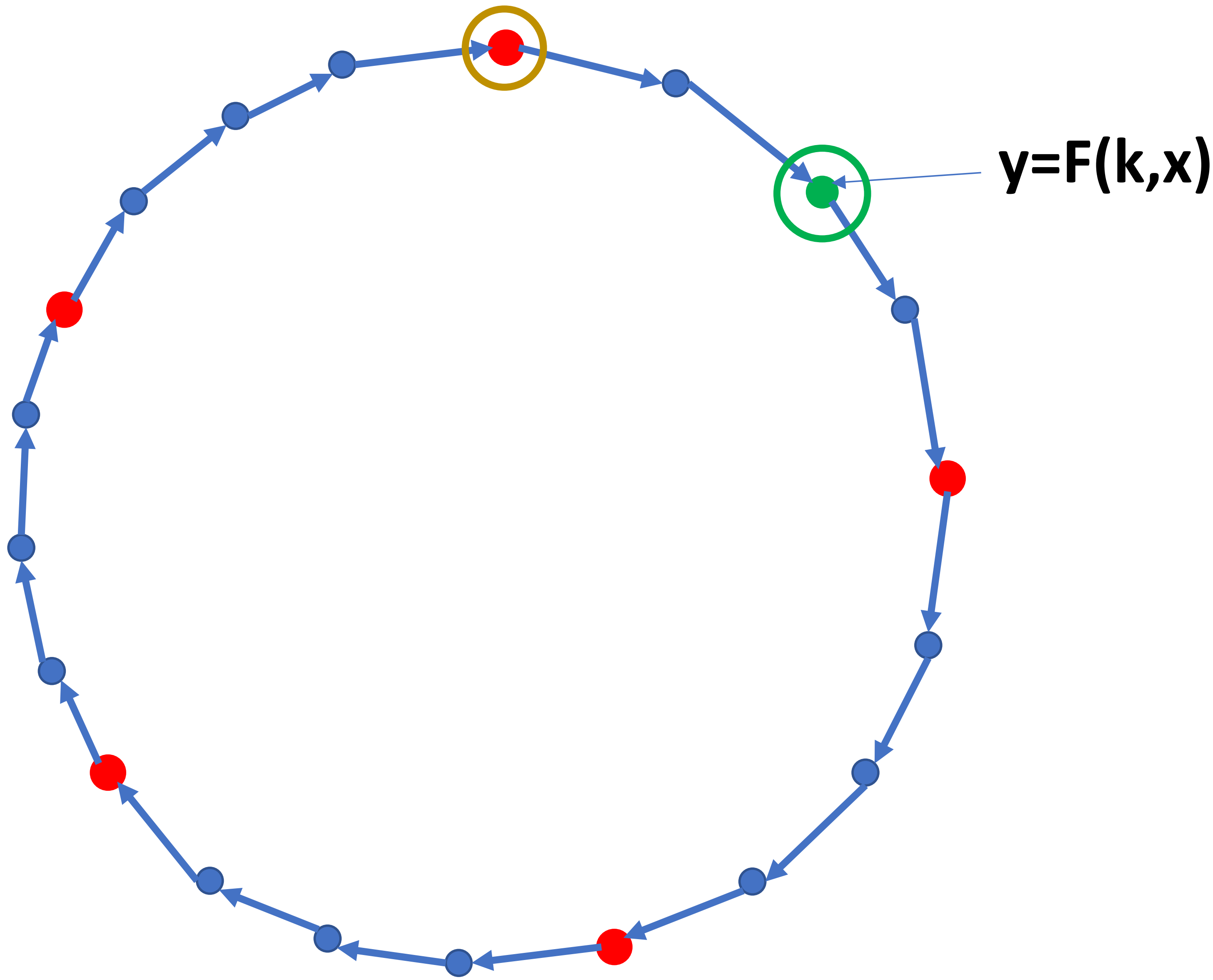
Option 3: Hellman's Attack



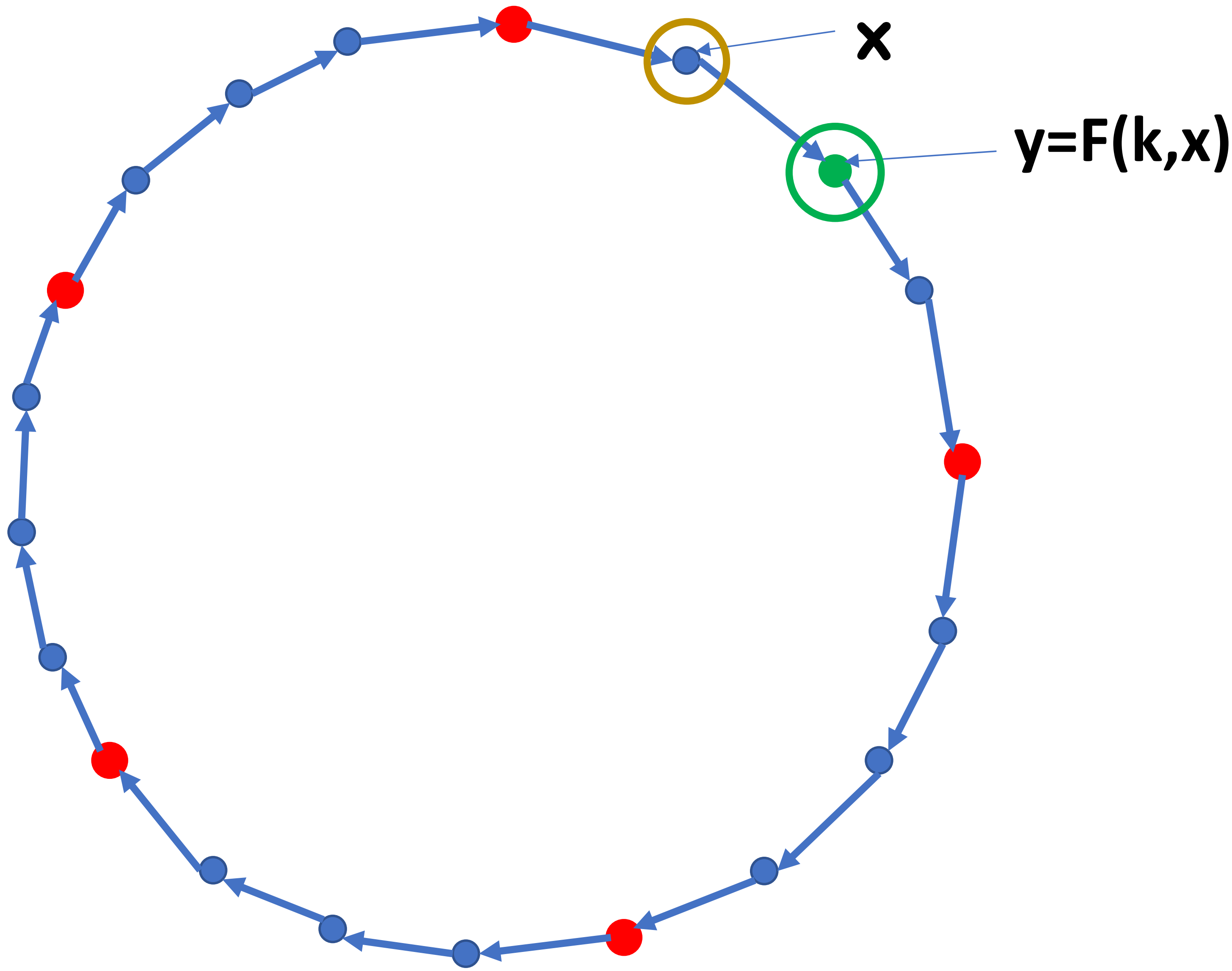
Option 3: Hellman's Attack



Option 3: Hellman's Attack



Option 3: Hellman's Attack



Option 3: Hellman's Attack

Preprocessing Time: $N=2^n$
Space: N/t
Online Time: t

Time-space tradeoff: **space × online time ≈** N

For non-cycles, attack is a bit harder, but nonetheless possible

Related Key Attacks

Properly designed crypto will always use random, independent keys for every application

However, sometimes people don't follow the rules

Related key attack: have messages encrypted under similar keys

(Recall RC4 used for encryption, **RC4(IV,k)**)

For AES 256, can attack in 2^{110} space/time

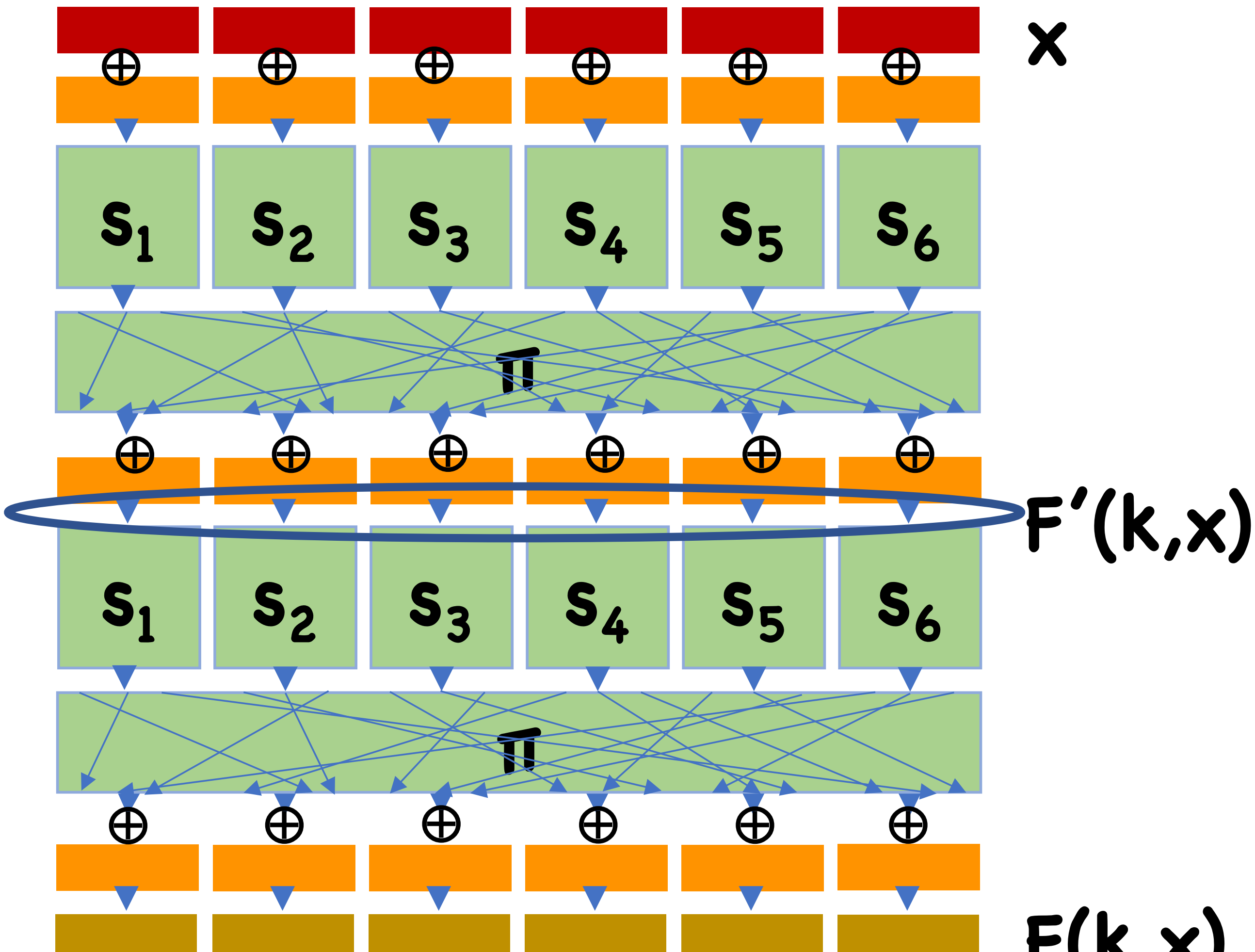
Differential Cryptanalysis

Suppose there were Δ_x,Δ_z such that, for random key k and random x_1,x_2 such that $x_1 \oplus x_2 = \Delta_x$, $F(k,x_1)\oplus F(k,x_2)=\Delta_z$ with probability $p \gg 2^{-n}$

- Call (Δ_x,Δ_z) a differential
- p is probability of differential
- $\approx 2^{-n}$ is probability of differential for random permutation

Yields distinguishing attack. With some effort, can also recover secret key

Differential Cryptanalysis

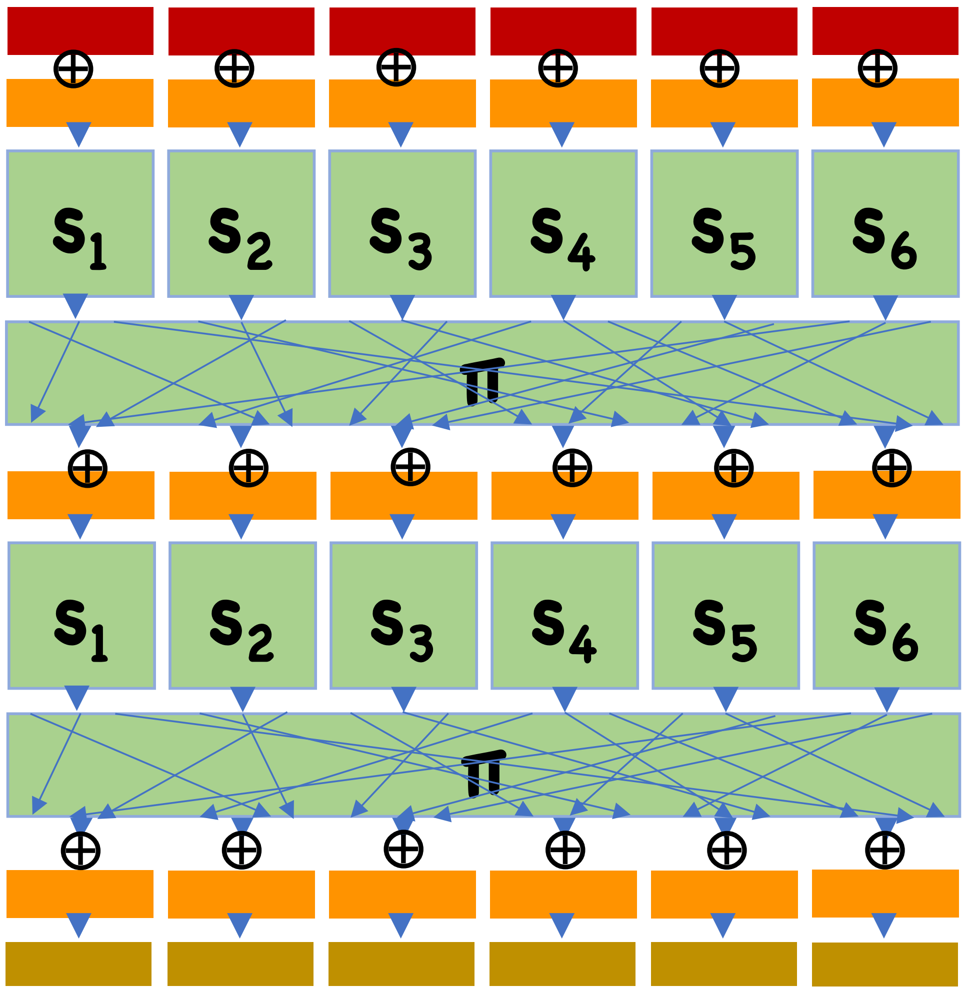


Differential Cryptanalysis

Attack:

- Suppose we have differential (Δ_x, Δ_y) for F'
- Choose many random pairs (x_1, x_2) s.t. $x_1 \oplus x_2 = \Delta_x$
- Make queries on each x_1, x_2, obtaining y_1, y_2
- Guess final round key k_r',
  - Use differentials to determine if guess was correct

Differential Cryptanalysis



Guess $\mathbf{k}_r'$

(x_0^1, x_1^1), (x_0^2, x_1^2), (x_0^3, x_1^3), (x_0^4, x_1^4), …

PRP Oracle

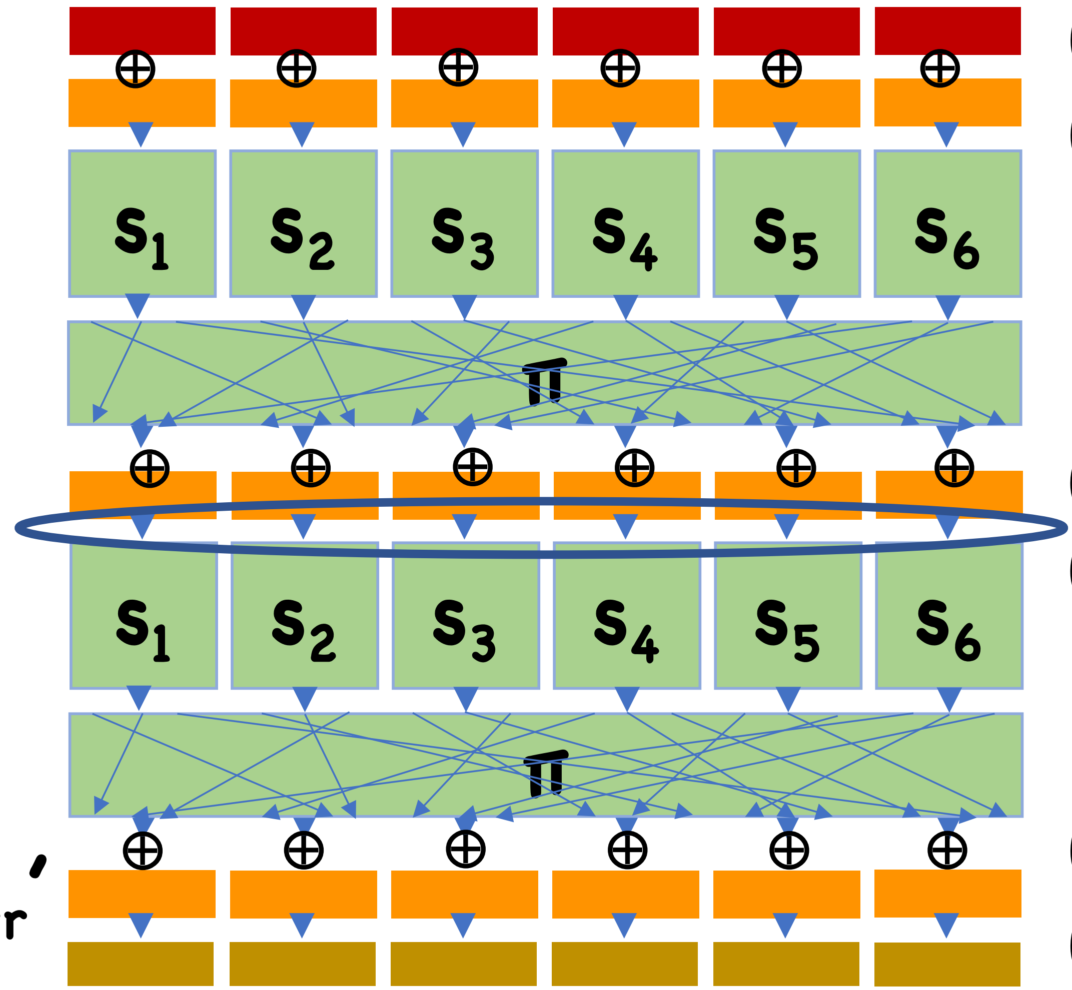
(y_0^1, y_1^1), (y_0^2, y_1^2), (y_0^3, y_1^3), (y_0^4, y_1^4), …

Differential Cryptanalysis

Attack:

- Choose many random pairs (x_1,x_2) s.t. $x_1 \oplus x_2 = \Delta_x$
- Make queries on each x_1, x_2, obtaining y_1,y_2
- For each possible final round key guess k_r',
    - Undo last round assuming k_r', obtaining (z_1',z_2')
    - Look for $z_1' \oplus z_2' = \Delta_z$
    - If right guess, expect $\approx$ p fraction
    - If wrong guess, expect $\approx$ 2^{-n} fraction

Differential Cryptanalysis



(x_0^1, x_1^1), (x_0^2, x_1^2), (x_0^3, x_1^3), (x_0^4, x_1^4), …

(z'^1_0, z'^1_1), (z'^2_0, z'^2_1), (z'^3_0, y'^3_1), (z'^4_0, z'^4_1), …

$\uparrow k_r'$

(y_0^1, y_1^1), (y_0^2, y_1^2), (y_0^3, y_1^3), (y_0^4, y_1^4), …

Guess $\mathbf{k_r}'$

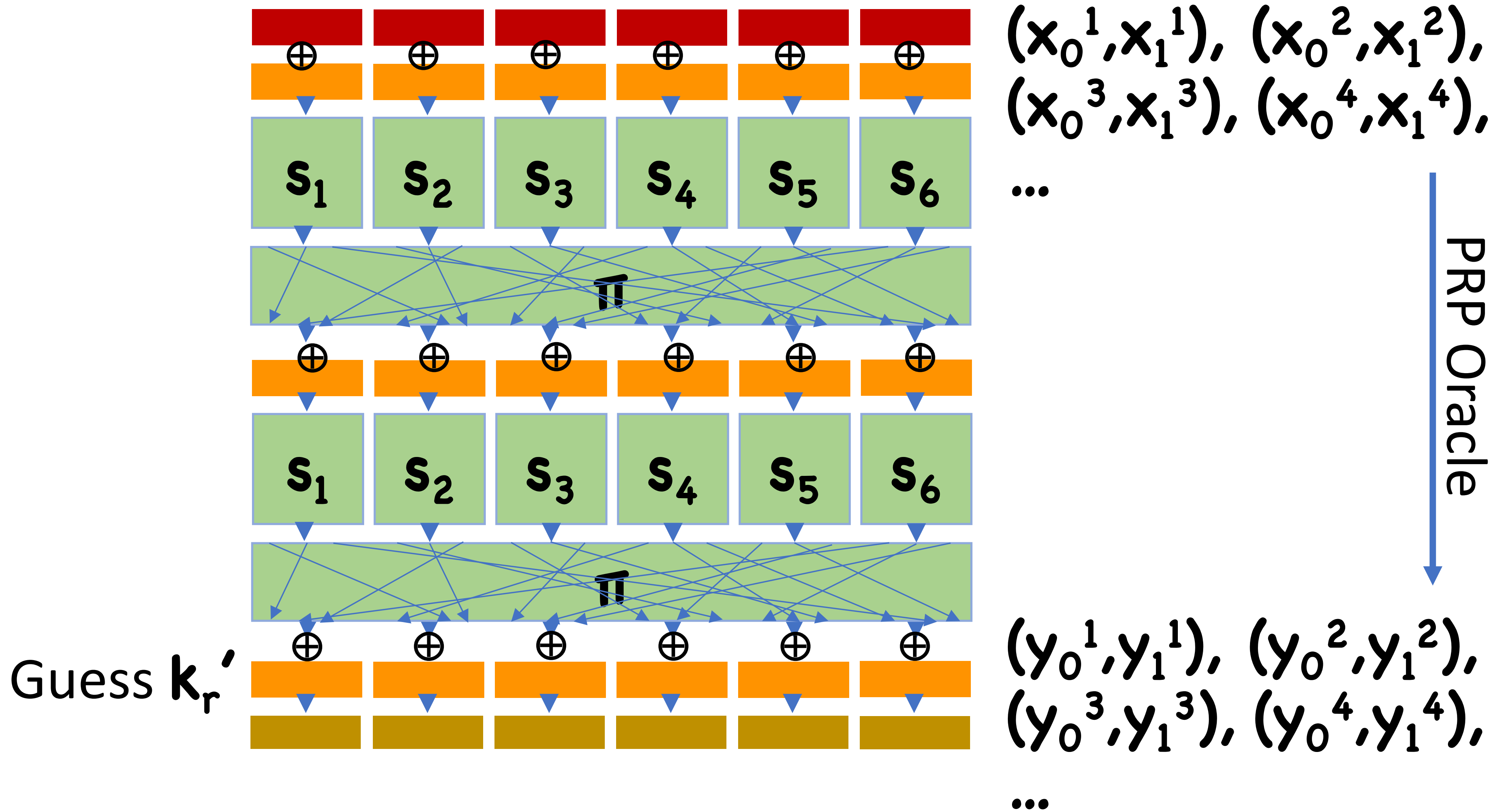
Differential Cryptanalysis

So far, inefficient since we have to iterate over all 2^n possible round keys

Instead, we can learn k_r 8 bits at a time

- Guess 8 bits of k_r at a time
- Iterate through all 2^8 possible values for those 8 bits
    - Compute 8 bits of z_1', z_2', look for (portion of) differential
- Which bits to choose?

Differential Cryptanalysis



Differential Cryptanalysis

Extending to further levels:

- One $\mathbf{k_r}$ is known, can un-compute last layer
- Now perform same attack on round-reduced cipher
- Repeat until all round keys have been found

Finding Differentials

So far, assumed differential given

How do we find it?
- Can’t simply brute force all possible differentials

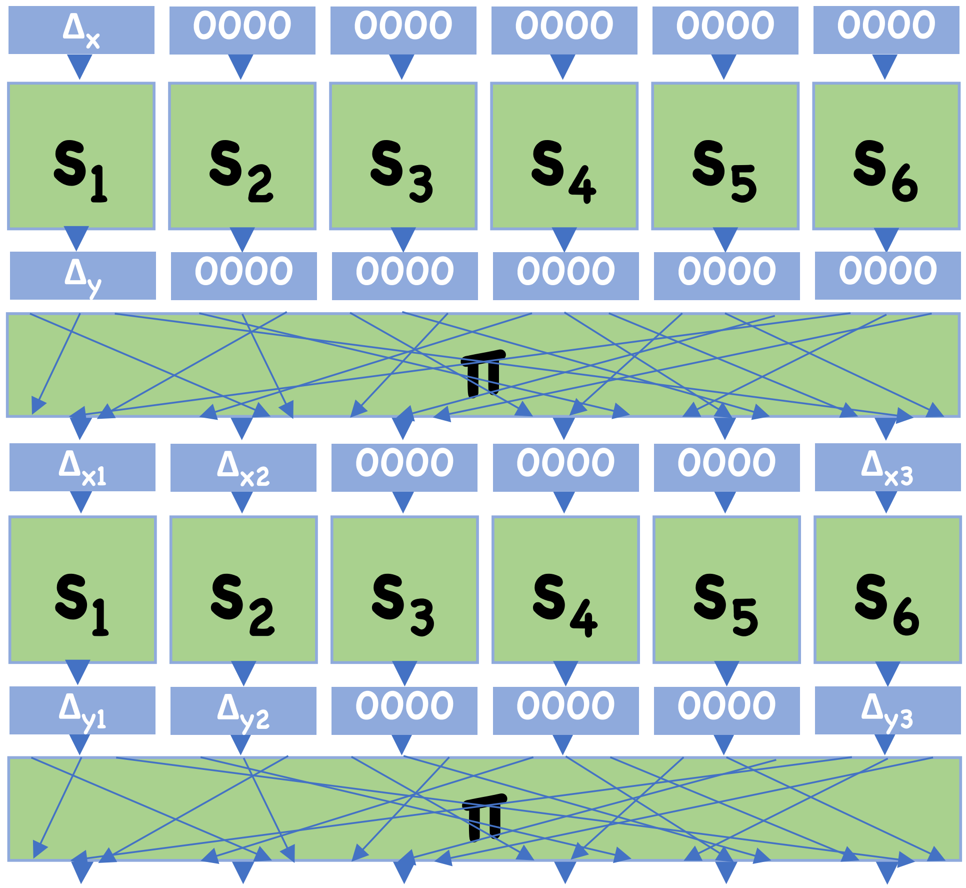
Finding Differentials

Solution: look for differentials in S-boxes

- Only 2^8 possible differences, so we can actually look for all possible differentials

- Then trace differentials through the evaluation
  - Key mixing does not affect differentials
  - Diffusion steps just shuffle differential bits

Differential Cryptanalysis



...

Differential Cryptanalysis in Practice

Used to attack real ciphers

- FEAL-8, proposed as alternative to DES in 1987
  - requires just 1000 chosen input/output pairs, 2 minutes computation time in 1990's

- Also theoretical attacks on DES
  - Requires 2^{47} chosen input/output pairs
  - Very difficult to obtain in real world applications
  - Therefore, DES is still considered relatively secure
  - Small changes to S-boxes in DES lead to much better differential attacks

Linear Cryptanalysis

High level idea: look for linear relationships that hold with too-high a probability

- E.g. $x_1 \oplus x_5 \oplus x_{17} \oplus y_3 \oplus y_6 \oplus y_{12} \oplus y_{21} = 0$

Can show that if happen with too-high probability, can completely recover key

Important feature: only requires *known* plaintext as opposed to *chosen* plaintext

- Much easier to carry out in practice
- Ex: DES can be broken with 2^{43} input/output pairs

Block Cipher Design

S-boxes are designed to minimize differential and linear cryptanalysis

- Cannot completely remove differentials/linear relations, but can minimize their probability

Increasing number of rounds helps

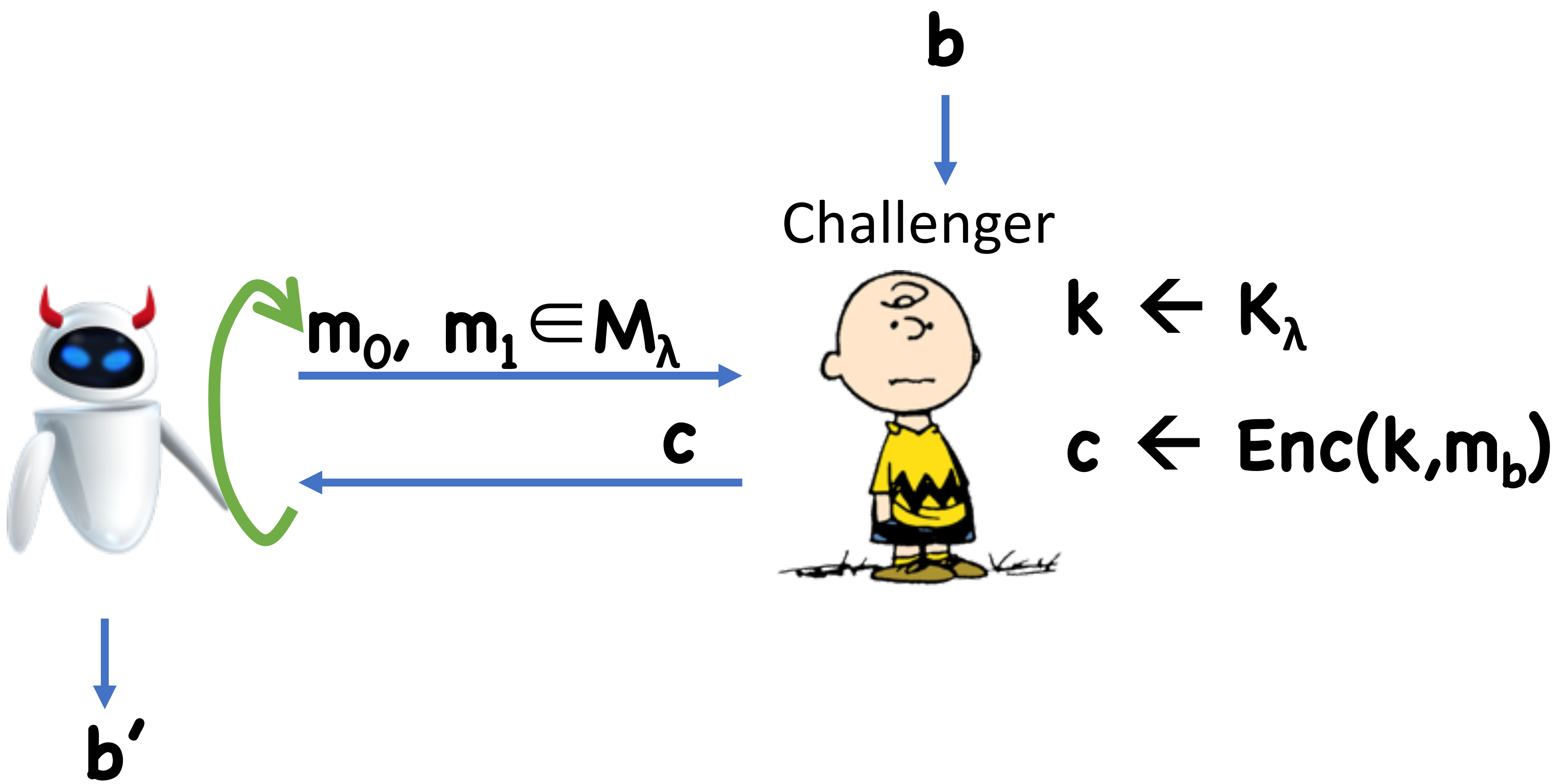
- Likelihood of differential decreases each round

Holiwudd Criptoe!

Device is top of the line.
AES cipher locks, brute force decryption is the only way.... It's effective, but slow. Very slow.

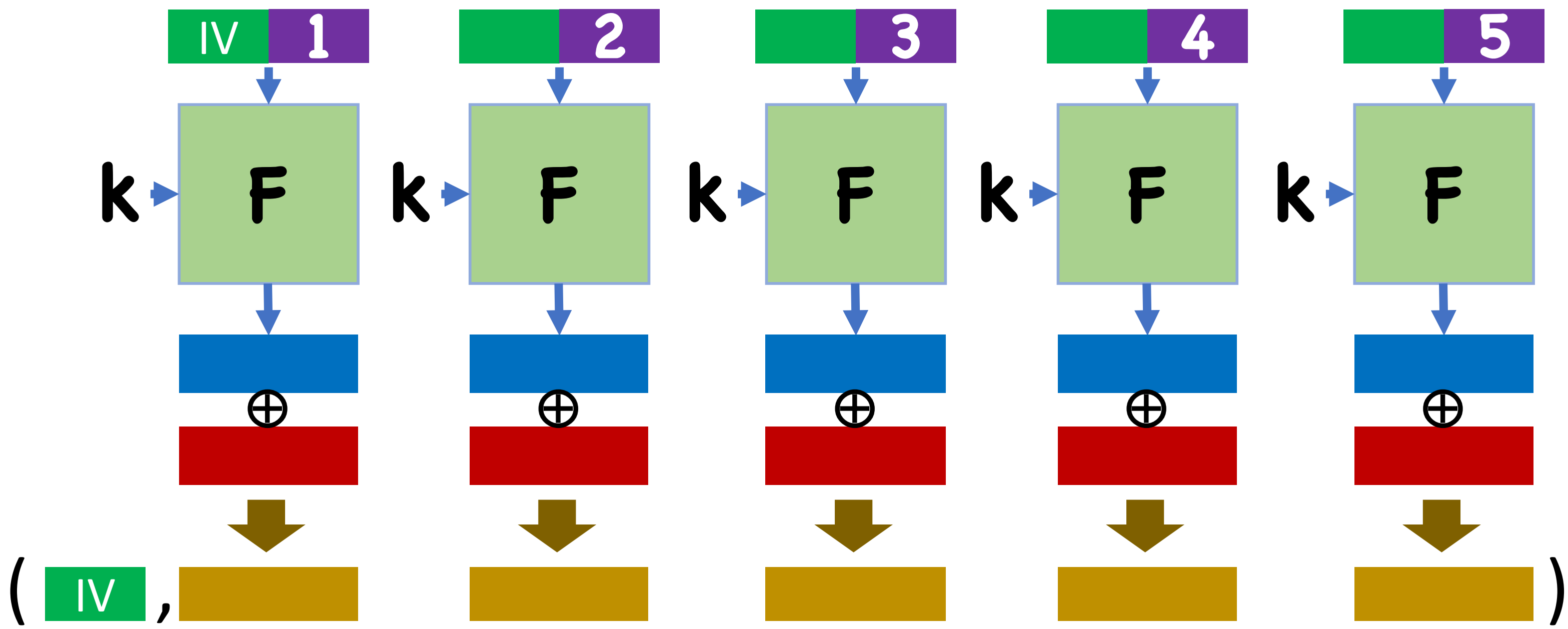
Message Integrity

Recall: CPA Security



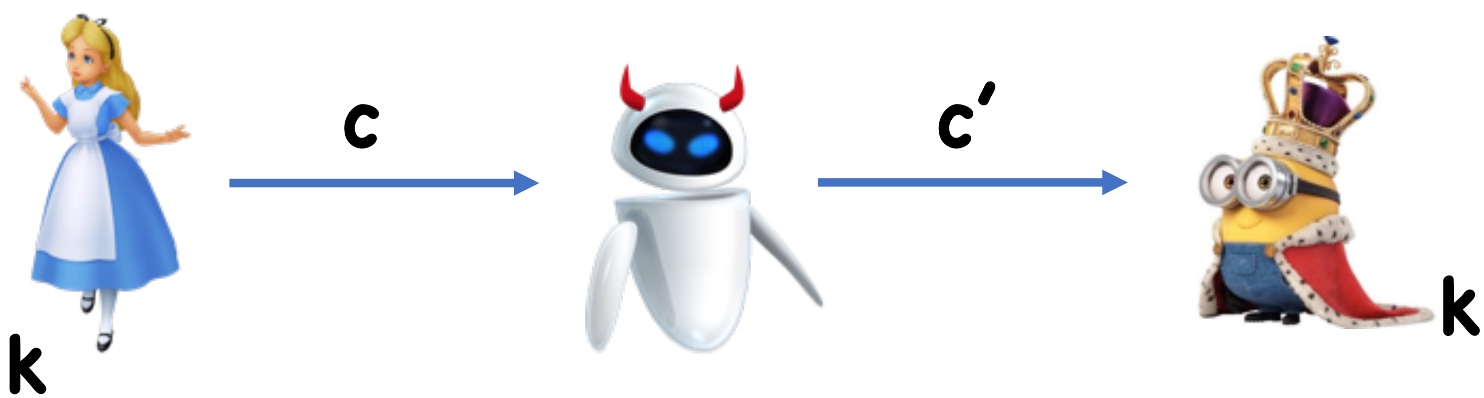
$\text{LoR-Exp}_b($  $, \lambda)$

Recall: Counter Mode (CTR)



Limitations of CPA security

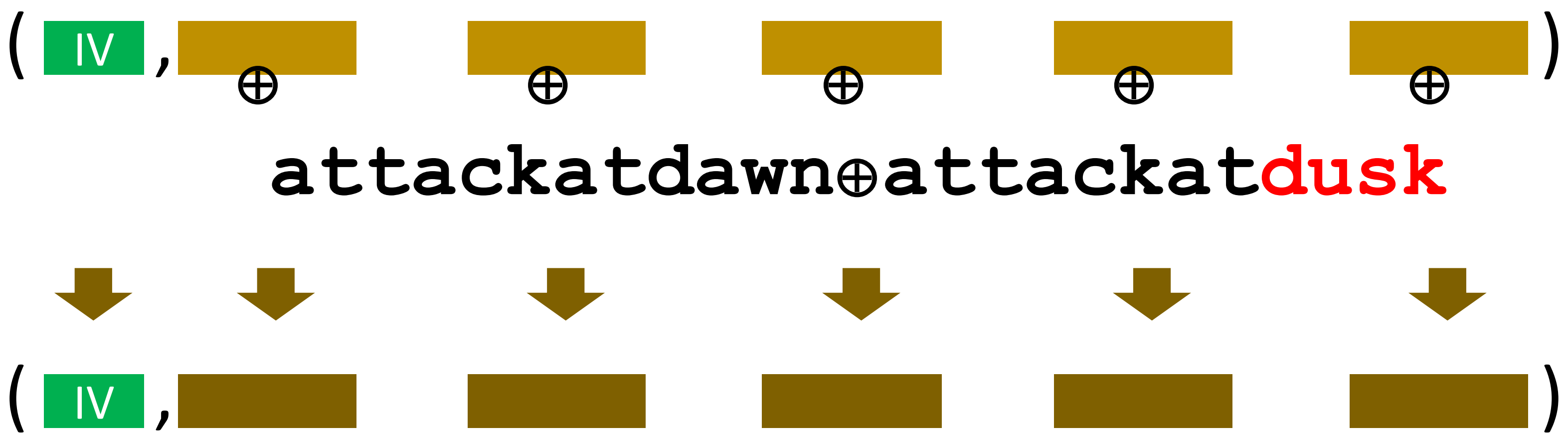
attackatdawn



attackat**dusk**

How?

Limitations of CPA Security



Malleability

Some encryption schemes are *malleable*

- Can modify ciphertext to cause predictable changes to plaintext

Examples: basically everything we’ve seen so far

- Stream ciphers
- CTR
- CBC
- ECB
- …

Message Integrity

We cannot stop adversary from changing the message in route to Bob

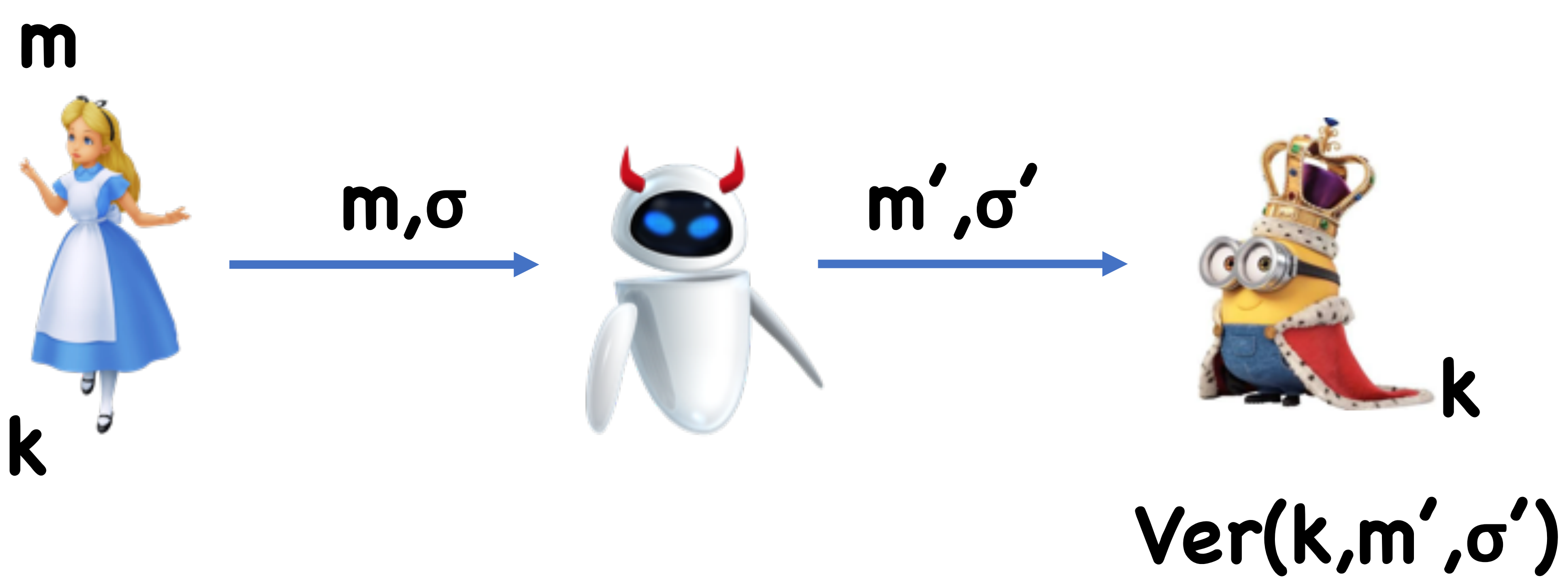
However, we can hope to have Bob perform some check on the message he receives to ensure it was sent by Alice and not modified

- If check fails, Bob rejects the message

For now, we won’t care about message secrecy

- We will add it back in later

Message Authentication



Goal: If Eve changed **m**, Bob should reject

Message Authentication Codes

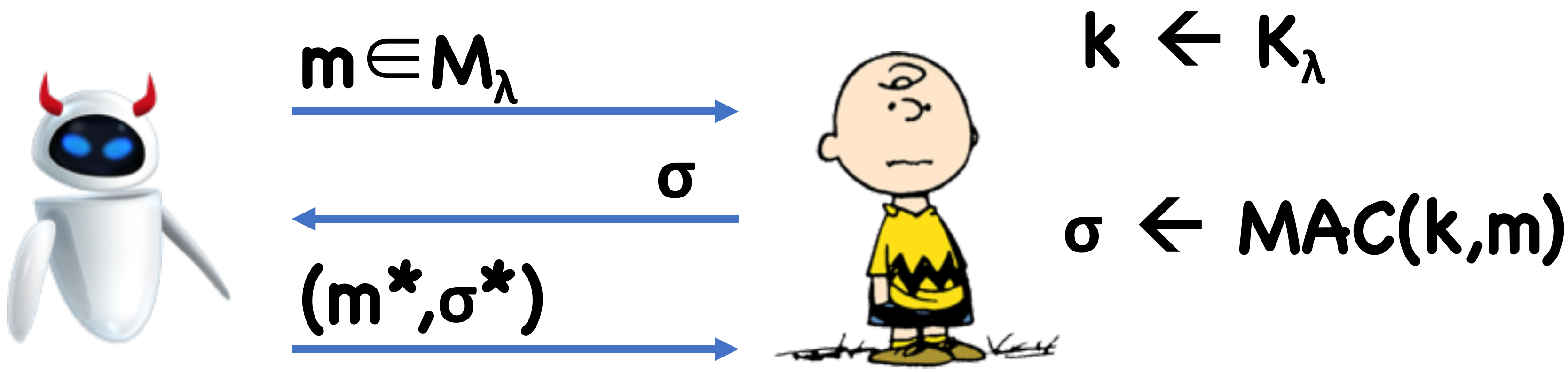
Syntax:

- Key space $\mathbf{K}_\lambda$
- Message space $\mathbf{M}_\lambda$
- Tag space $\mathbf{T}_\lambda$
- **MAC(k,m) → σ**
- **Ver(k,m,σ) → 0/1**

Correctness:

- $\forall$**m,k, Ver(k,m, MAC(k,m)) = 1**

1-time Security For MACs



Output 1 iff:

- $m^* \neq m$
- $Ver(k,m^*,\sigma^*) = 1$

1CMA-Adv( , λ) = Pr[ outputs 1]

Definition: (MAC,Ver) is 1-time statistically secure under a chosen message attack (**statistically 1CMA-secure)** if, for all , $\exists$ negligible **ε** such that:

1CMA-Adv(**, λ) ≤ ε(λ)**

Announcements/Reminders

Last day to submit HW2

- Submit through Gradescope

PR1 Due October 6

Heads up: HW3 will be out soon, due on Oct 20