Blockcipher Security Notions

Martijn Stam

Department of Computer Science,
University Of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB
United Kingdom.

Šibenik, 7 June 2016

Data Encryption Standard (DES)

An Early Blockcipher

1970s: DES provided the first FIPS standard for a blockcipher

It takes as input:

- a 56-bit string k called the key
- a 64-bit string x called the plaintext or input block.

and outputs

- a 64-bit string y called the ciphertext or output block.

The algorithm is stateless, deterministic, and invertible.

$\forall_{k,x}$ If $y \leftarrow \mathrm{DES}_k(x)$ then $x \leftarrow \mathrm{DES}_k^{-1}(y)$

Advanced Encryption Standard (AES)

A Modern Blockcipher

Turn of Century: NIST approves AES as successor of DES.

AES-**128** takes as input:

- a **128**-bit string k called the key
- a 128-bit string x called the plaintext or input block.

and outputs

- a 128-bit string y called the ciphertext or output block.

The algorithm is stateless, deterministic, and invertible.

$\forall_{k,x}$ If $y \leftarrow \mathrm{AES}_k(x)$ then $x \leftarrow \mathrm{AES}_k^{-1}(y)$

Advanced Encryption Standard (AES)

A Modern Blockcipher

Turn of Century: NIST approves AES as successor of DES.

AES-**192** takes as input:

- a **192**-bit string k called the key
- a 128-bit string x called the plaintext or input block.

and outputs

- a 128-bit string y called the ciphertext or output block.

The algorithm is stateless, deterministic, and invertible.

$\forall_{k,x}$ If $y \leftarrow \mathrm{AES}_k(x)$ then $x \leftarrow \mathrm{AES}_k^{-1}(y)$

Advanced Encryption Standard (AES)

A Modern Blockcipher

Turn of Century: NIST approves AES as successor of DES.

AES-**256** takes as input:

- a **256**-bit string k called the key
- a 128-bit string x called the plaintext or input block.

and outputs

- a 128-bit string y called the ciphertext or output block.

The algorithm is stateless, deterministic, and invertible.

$\forall_{k,x}$ If $y \leftarrow \mathrm{AES}_k(x)$ then $x \leftarrow \mathrm{AES}_k^{-1}(y)$

Blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K}$ is the set of keys,
- $\mathcal{X}$ the set of plaintext blocks

Blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K}$ is the set of keys,
- $\mathcal{X}$ the set of plaintext blocks

Notation for blockciphers

- $\mathrm{Block}(\mathcal{K}, \mathcal{X})$ denotes the set of all possible blockciphers of given dimensions
- $\mathrm{Perm}(\mathcal{X})$ denotes the set of all permutations on $\mathcal{X}$.

Blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K}$ is the set of keys,
- $\mathcal{X}$ the set of plaintext blocks

Notation for $\mathsf{E} \in \mathrm{Block}(\mathcal{K}, \mathcal{X})$

Let $\mathsf{k} \in \mathcal{K}$ we write $\mathsf{E}_\mathsf{k}(\cdot)$ for $\mathsf{E}(\mathsf{k}, \cdot)$.
As $\mathsf{E}_\mathsf{k} \in \mathrm{Perm}(\mathcal{X})$ it has an inverse $\mathsf{E}_\mathsf{k}^{-1}$ or D_k

Blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K}$ is the set of keys,
- $\mathcal{X}$ the set of plaintext blocks

Notation for $\mathsf{E} \in \mathrm{Block}(\mathcal{K}, \mathcal{X})$

Let $\mathsf{k} \in \mathcal{K}$ we write $\mathsf{E}_\mathsf{k}(\cdot)$ for $\mathsf{E}(\mathsf{k}, \cdot)$.
As $\mathsf{E}_\mathsf{k} \in \mathrm{Perm}(\mathcal{X})$ it has an inverse $\mathsf{E}_\mathsf{k}^{-1}$ or D_k

For all $\mathsf{k} \in \mathcal{K}, x \in \mathcal{X}$:

$$\mathsf{D}_\mathsf{k}(\mathsf{E}_\mathsf{k}(x)) = \mathsf{E}_\mathsf{k}(\mathsf{D}_\mathsf{k}(x)) = x$$

Blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K}$ is the set of keys,
- $\mathcal{X}$ the set of plaintext blocks

Using bitstrings as inputs

- $\mathcal{K} = \{0,1\}^{\mathsf{K}}$ for some key-length $\mathsf{K} \in \mathbb{N}$
- $\mathcal{X} = \{0,1\}^{\mathsf{n}}$ for some block-length n.

Blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K}$ is the set of keys,
- $\mathcal{X}$ the set of plaintext blocks

Using bitstrings as inputs

- $\mathcal{K} = \{0,1\}^{\mathsf{K}}$ for some key-length $\mathsf{K} \in \mathbb{N}$
- $\mathcal{X} = \{0,1\}^{\mathsf{n}}$ for some block-length n.

- DES has $n = 64$ and $k = 56$;
- AES has $n = 128$ and $k \in \{128, 192, 256\}$

Blockcipher Security

Ideas?

What security would you expect from a blockcipher?

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key!

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key!
- learn plaintexts!

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key!
- learn plaintexts!
- predict ciphertexts!

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key!
- learn plaintexts!
- predict ciphertexts!
- distinguish its output from random!

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key! **But when?**
- learn plaintexts!
- predict ciphertexts!
- distinguish its output from random!

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key! **But when?**
- learn plaintexts! **Which plaintexts?**
- predict ciphertexts!
- distinguish its output from random!

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key! **But when?**
- learn plaintexts! **Which plaintexts?**
- predict ciphertexts! **In what context?**
- distinguish its output from random!

Blockcipher Security

Ideas?

Some random thoughts...

It should be hard to

- recover they key! **But when?**
- learn plaintexts! **Which plaintexts?**
- predict ciphertexts! **In what context?**
- distinguish its output from random! **Random in what sense?**

Blockcipher Security

Ideas?

More precise definitions are needed, that

- highlight what an adversary can do and tries to achieve
- take into account the context in which the blockcipher is used

Blockcipher Security

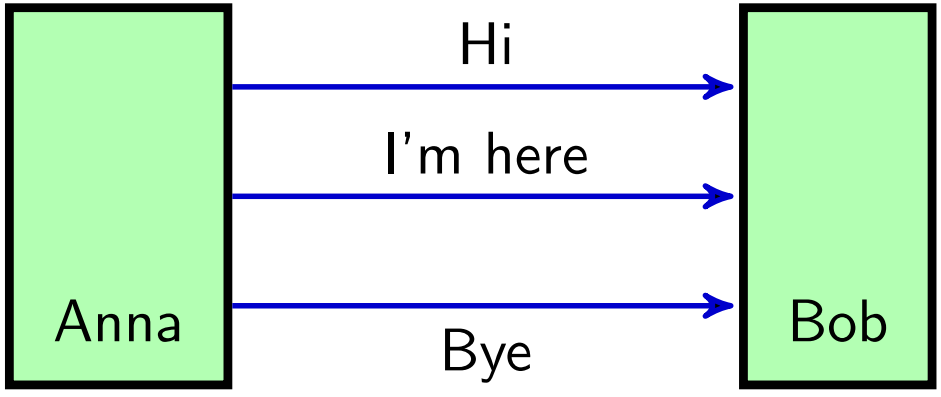
Ideas?

More precise definitions are needed, that

- highlight what an adversary can do and tries to achieve
- take into account the context in which the blockcipher is used
- ...so useful conclusions for real world applications can be drawn.

How are blockciphers used?

Scenario 1: Secure Communication

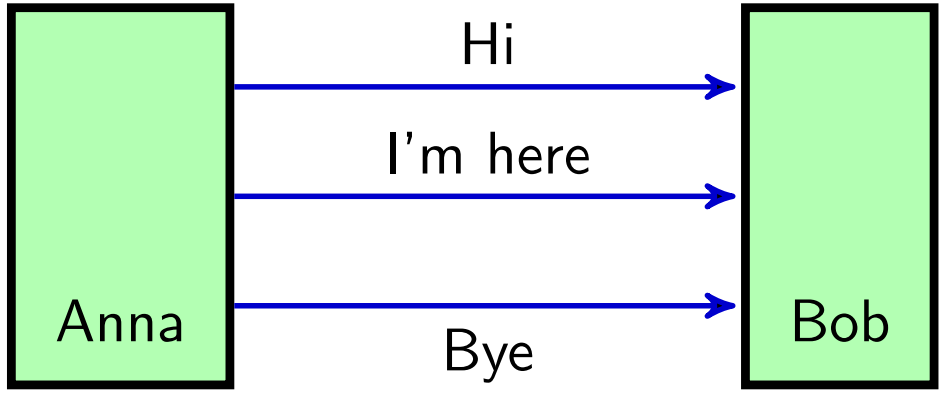


Two parties, Anna and Bob want to communicate with each other:

- Anna wants to send Bob messages;

How are blockciphers used?

Scenario 1: Secure Communication

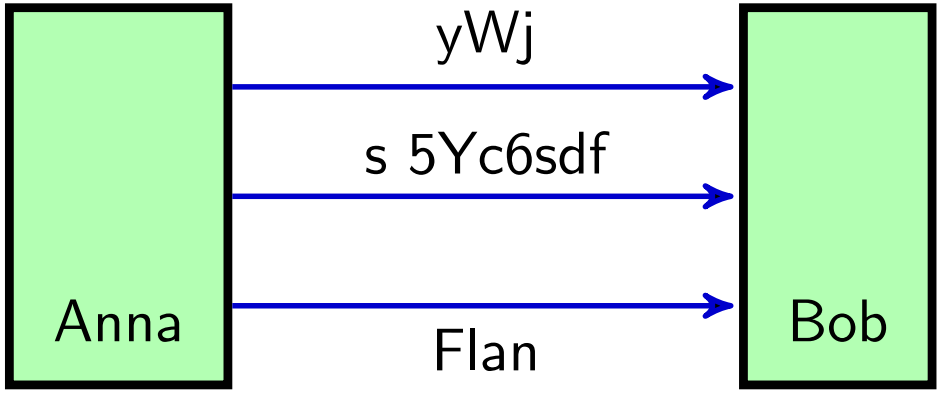


Two parties, Anna and Bob want to communicate with each other:

- Anna wants to send Bob messages;
- The content of the messages should remain hidden from Eve;

How are blockciphers used?

Scenario 1: Secure Communication

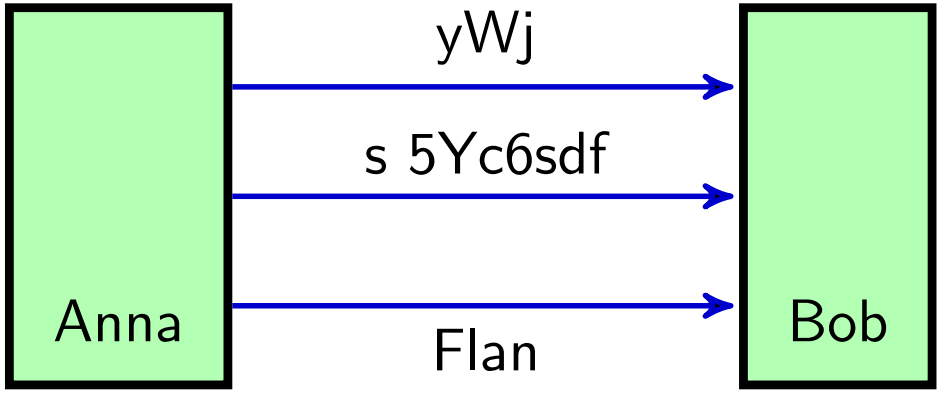


Two parties, Anna and Bob want to communicate with each other:

- Anna wants to send Bob messages;
- The content of the messages should remain hidden from Eve;

How are blockciphers used?

Scenario 1: Secure Communication

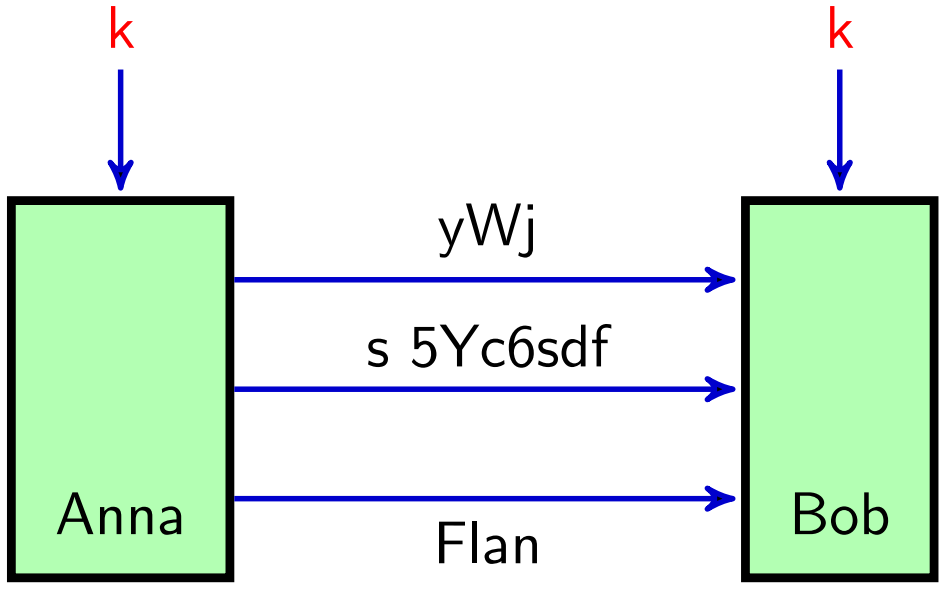


Two parties, Anna and Bob want to communicate with each other:

- Anna wants to send Bob messages;
- The content of the messages should remain hidden from Eve;
- Adversary Eve can see but not modify the transmissions.

How are blockciphers used?

Scenario 1: Secure Communication

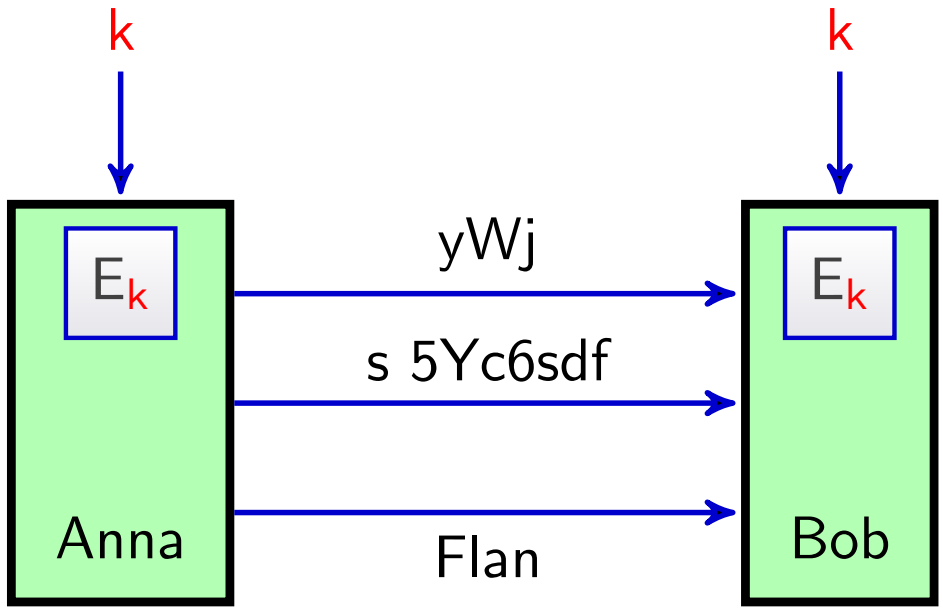


Some enabling assumptions:

- Anna and Bob already magically share a secret key;

How are blockciphers used?

Scenario 1: Secure Communication

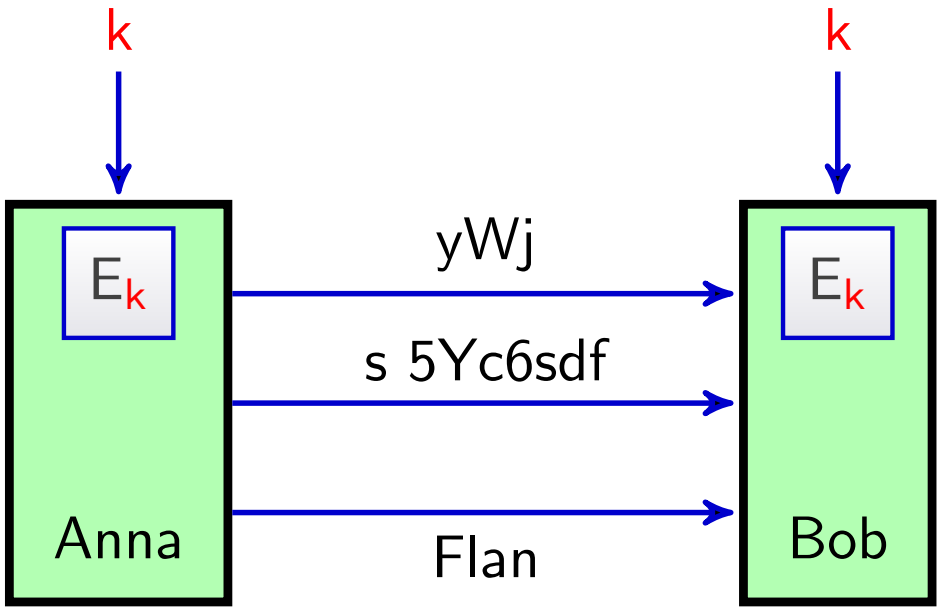


Some enabling assumptions:

- Anna and Bob already magically share a secret key;
- They both like the same blockcipher

How are blockciphers used?

Scenario 1: Secure Communication

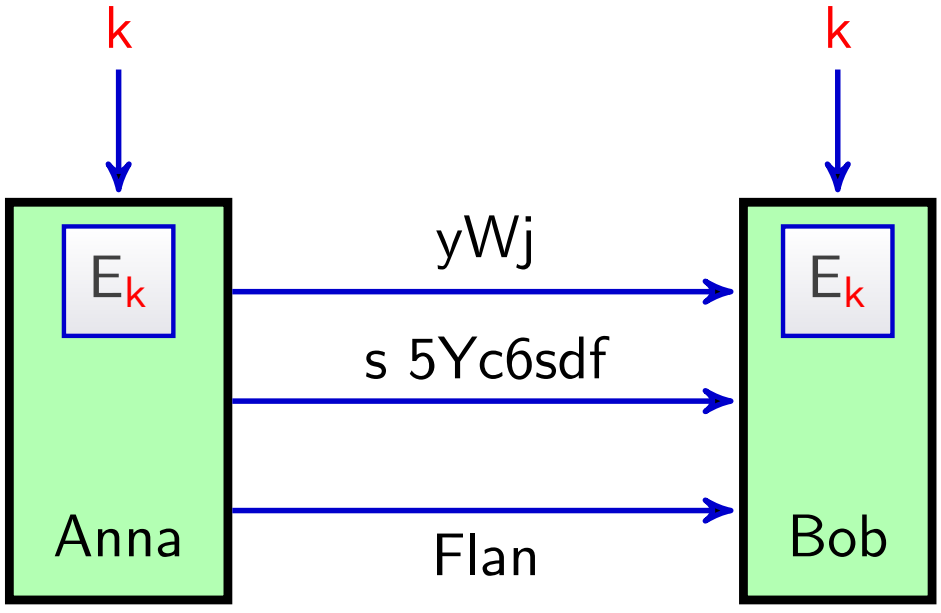


Some enabling assumptions:

- Anna and Bob already magically share a secret key;
- They both like the same blockcipher
- Anna swims in a pool

How are blockciphers used?

Scenario 1: Secure Communication

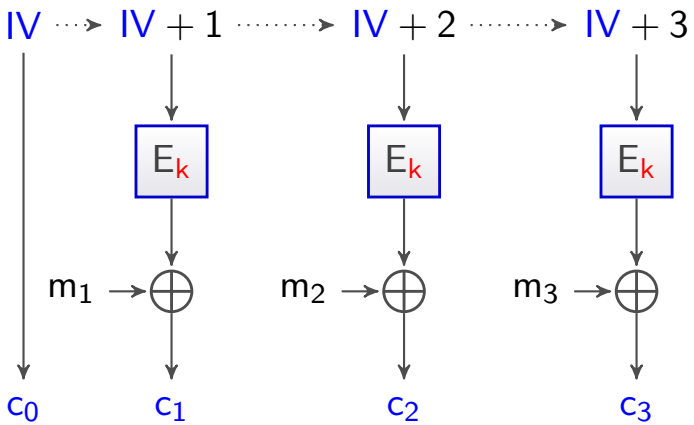


Some enabling assumptions:

- Anna and Bob already magically share a secret key;
- They both like the same blockcipher
- Anna swims in a pool of randomness

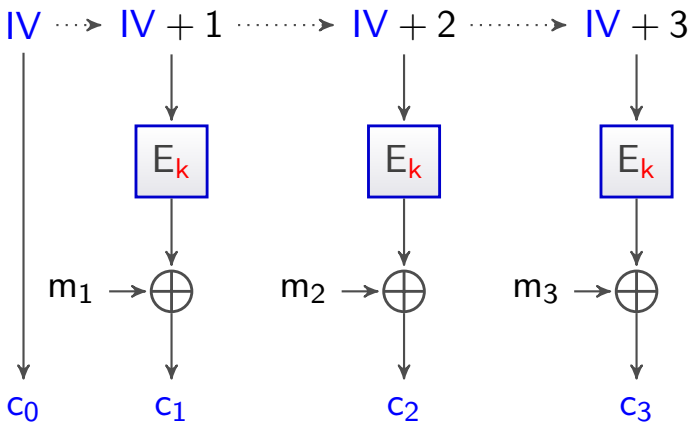
Confidentiality of a single 3-block message

CTR Encryption



Confidentiality of a single 3-block message

CTR Encryption



Game (informally):

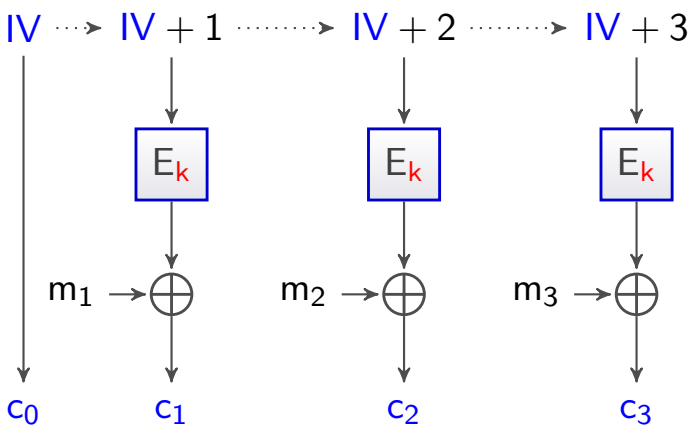
Adversary picks (m_1, m_2, m_3)

Receives $(c_0, \cdots, c_3)$ which is either true encryption or random.

Needs to guess which.

Confidentiality of a single 3-block message

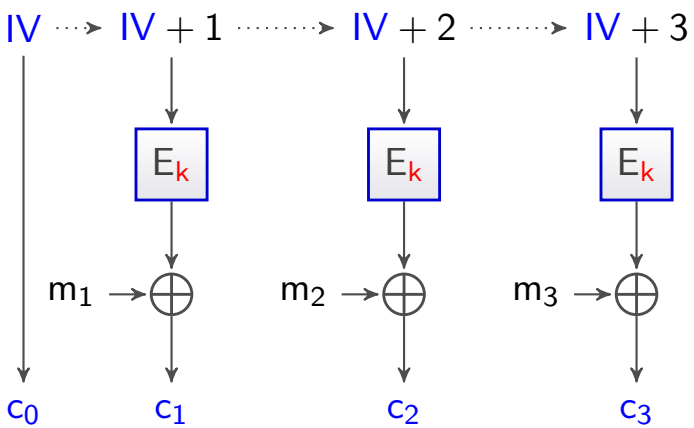
CTR Encryption



- Random secret key k is used repeatedly
- Blockcipher inputs certainly not jointly random

Confidentiality of a single 3-block message

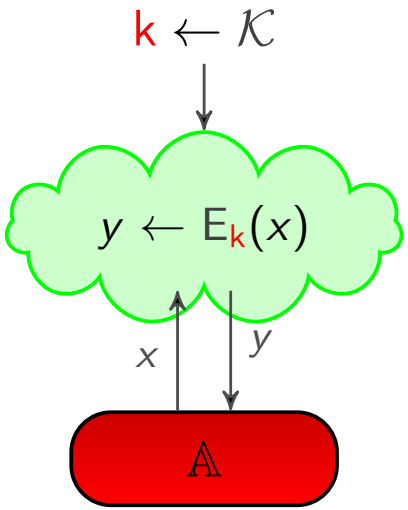
CTR Encryption



- Random secret key k is used repeatedly
- Blockcipher inputs certainly not jointly random
- Easiest to give adversary full control over them!

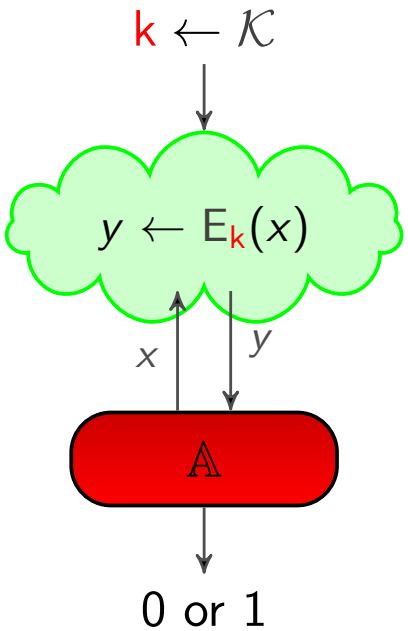
Blockcipher Security

Pseudorandom Permutations



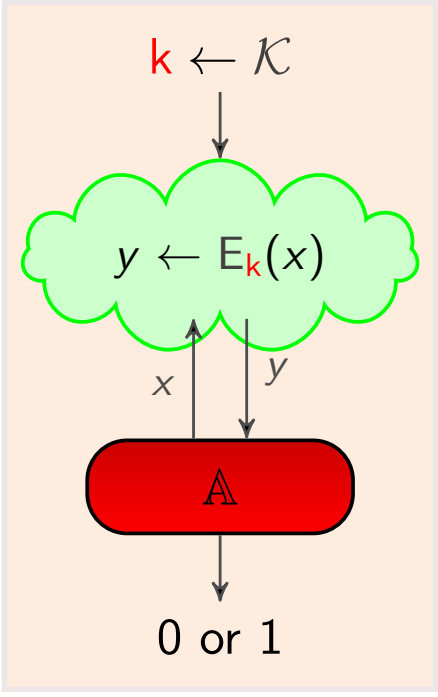
Blockcipher Security

Pseudorandom Permutations



Blockcipher Security

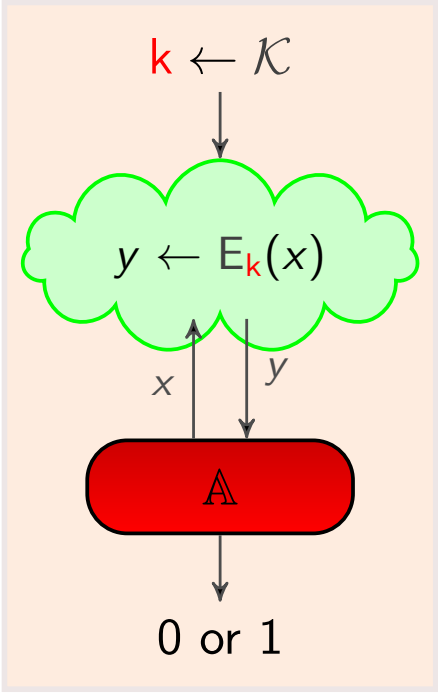
Pseudorandom Permutations



Real blockcipher world

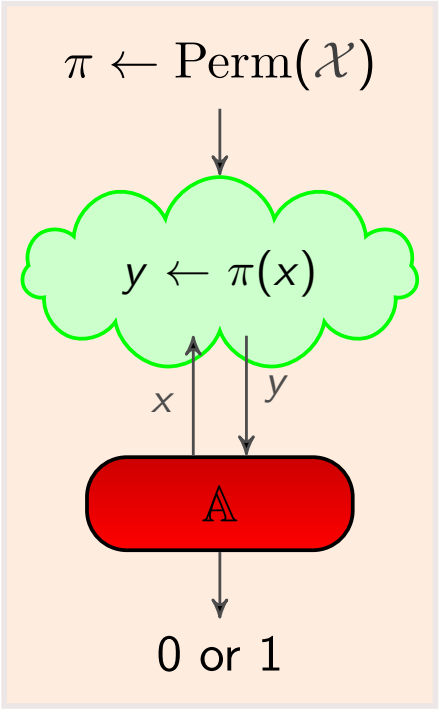
Blockcipher Security

Pseudorandom Permutations



Real blockcipher world

v



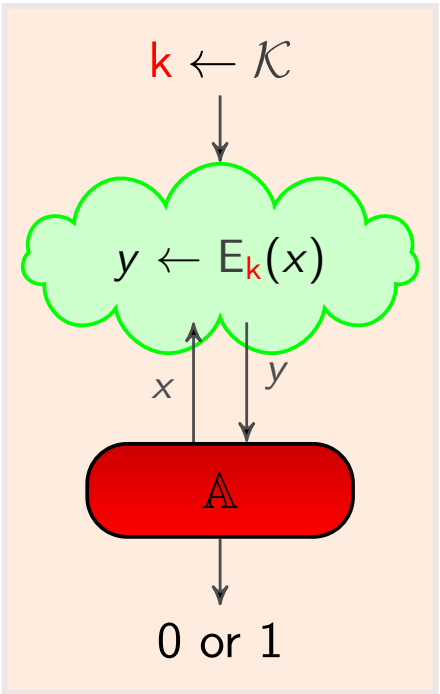
Random permutation world

How well can an adversary distinguish?

Blockcipher Security

Pseudorandom Permutations

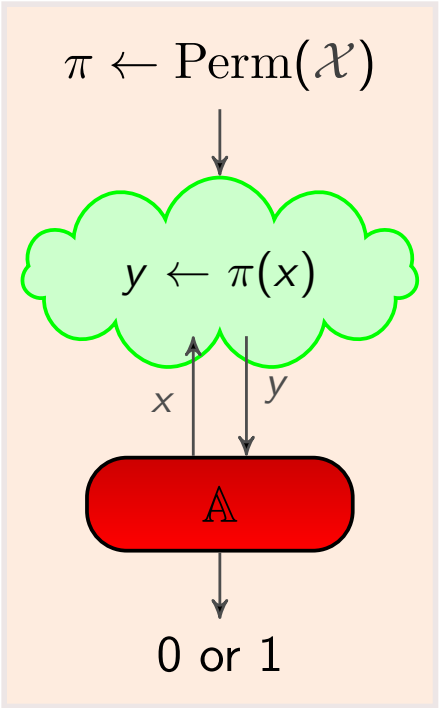
$\mathbf{Exp}_{\mathsf{E}}^{prp\text{-}0}(\mathbb{A})$



Real blockcipher world

v

$\mathbf{Exp}_{\mathsf{E}}^{prp\text{-}1}(\mathbb{A})$

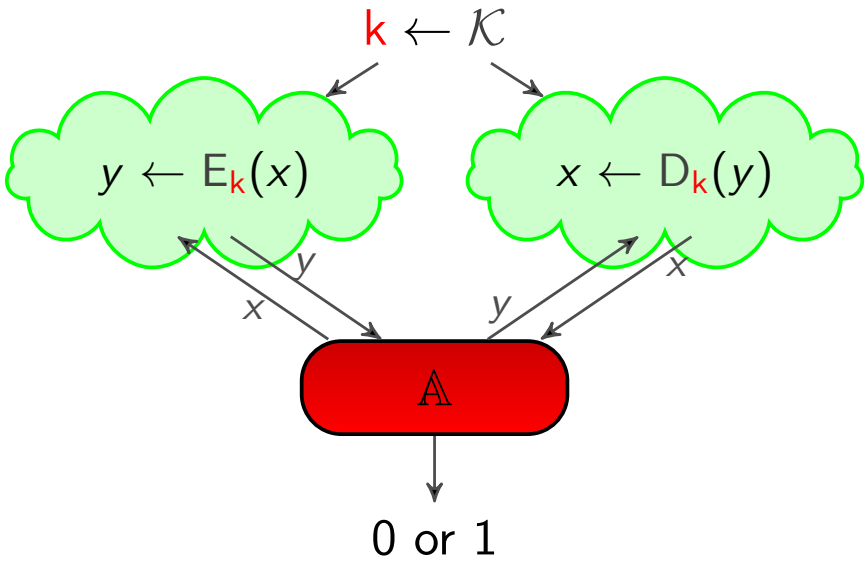


Random permutation world

$$\mathbf{Adv}_{\mathsf{E}}^{prp}(\mathbb{A}) = \left|\mathbf{Pr}\left[\mathbf{Exp}_{\mathsf{E}}^{prp\text{-}0}(\mathbb{A}) = 0\right] - \mathbf{Pr}\left[\mathbf{Exp}_{\mathsf{E}}^{prp\text{-}1}(\mathbb{A}) = 0\right]\right|$$

Blockcipher Security

Strong Pseudorandom Permutations

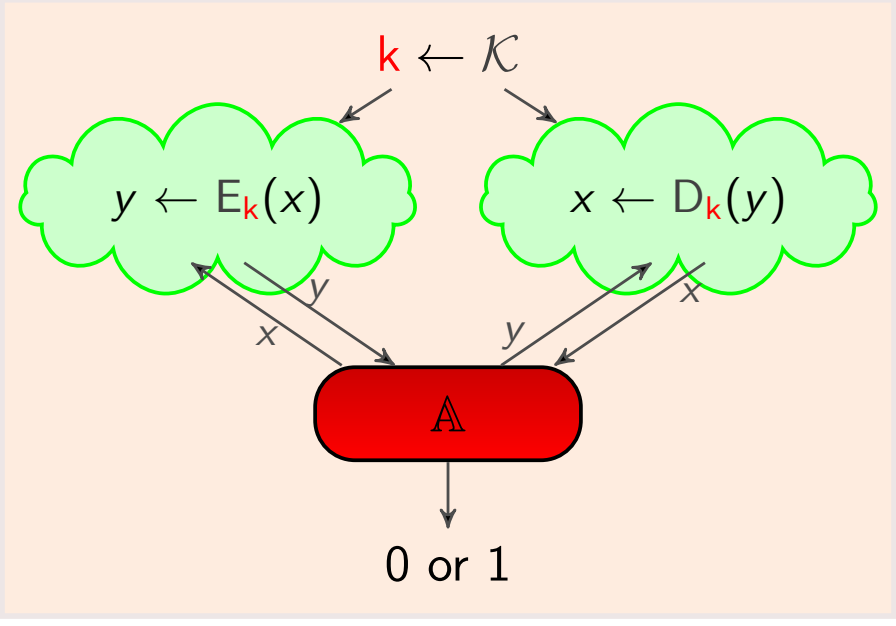


Strong security: Adversary has access to inverse as well

Blockcipher Security

Strong Pseudorandom Permutations

$\mathbf{Exp}_{\mathsf{E}}^{sprp\text{-}0}(\mathbb{A})$



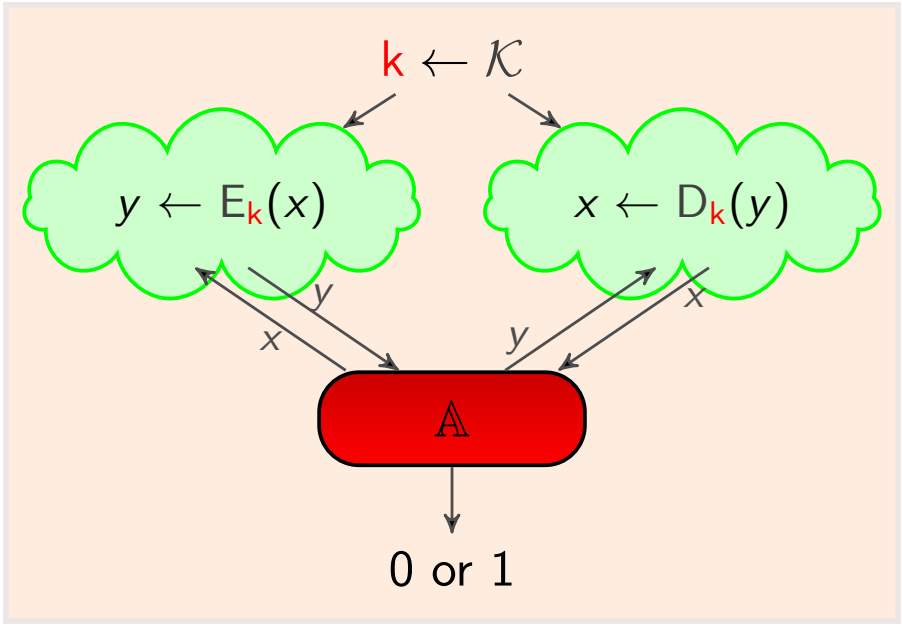
Real blockcipher world

Strong security: Adversary has access to inverse as well

Blockcipher Security

Strong Pseudorandom Permutations

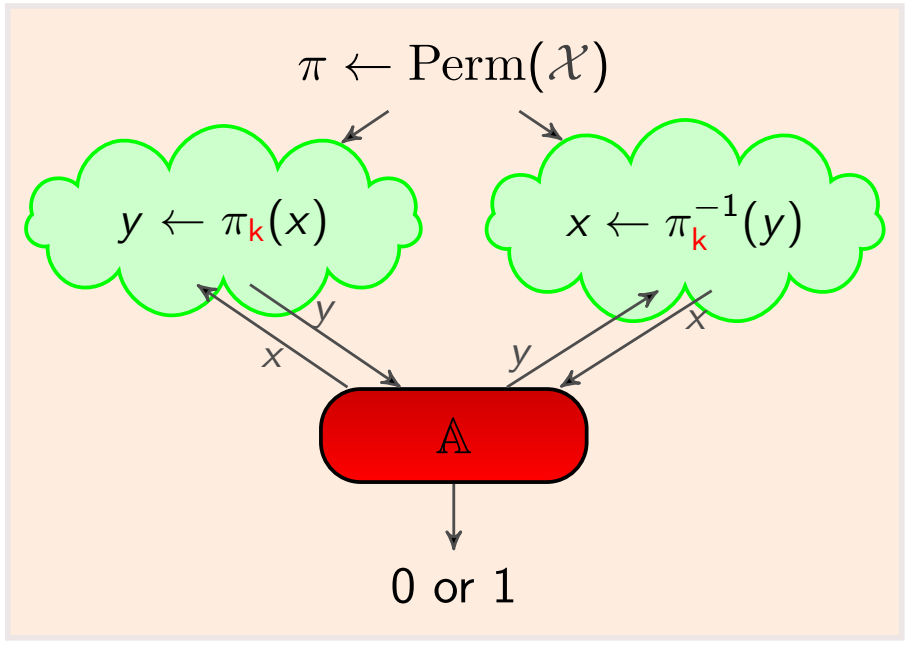
$\mathbf{Exp}_{\mathsf{E}}^{sprp\text{-}0}(\mathbb{A})$



Real blockcipher world

v

$\mathbf{Exp}_{\mathsf{E}}^{sprp\text{-}1}(\mathbb{A})$

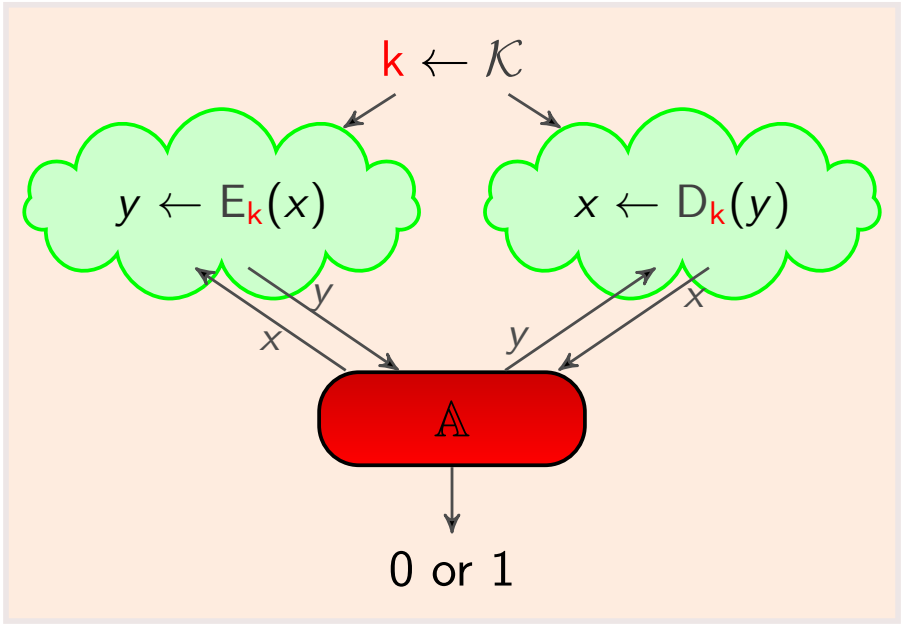


Random permutation world

Blockcipher Security

Strong Pseudorandom Permutations

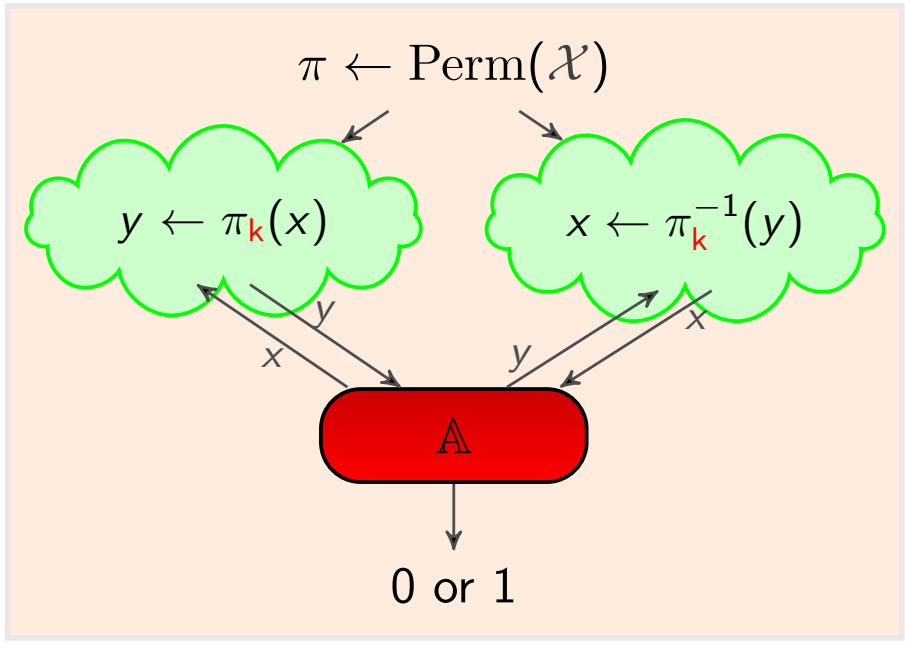
$\mathbf{Exp}_{\mathsf{E}}^{sprp\text{-}0}(\mathbb{A})$



Real blockcipher world

v

$\mathbf{Exp}_{\mathsf{E}}^{sprp\text{-}1}(\mathbb{A})$



Random permutation world

$$\mathbf{Adv}_{\mathsf{E}}^{sprp}(\mathbb{A}) = \left|\mathbf{Pr}\left[\mathbf{Exp}_{\mathsf{E}}^{sprp\text{-}0}(\mathbb{A}) = 0\right] - \mathbf{Pr}\left[\mathbf{Exp}_{\mathsf{E}}^{sprp\text{-}1}(\mathbb{A}) = 0\right]\right|$$

(Strong) Pseudorandom Permutations

What it means

Implications 1

Security as a pseudorandom permutation implies

$\Rightarrow$ Key recovery under chosen plaintext attacks is hard

$\Rightarrow$ Some modes-of-operation can be proven secure

(Strong) Pseudorandom Permutations

What it means

Implications 2

Security as a **strong** pseudorandom permutation implies

- $\Rightarrow$ "Ordinary" pseudorandom permutation
- $\Rightarrow$ Key recovery under chosen **cipher**text attacks is hard
- $\Rightarrow$ Some **more** modes-of-operation can be proven secure

(Strong) Pseudorandom Permutations

What it means

Implications 2

Security as a **strong** pseudorandom permutation implies

$\Rightarrow$ "Ordinary" pseudorandom permutation

$\Rightarrow$ Key recovery under chosen **cipher**text attacks is hard

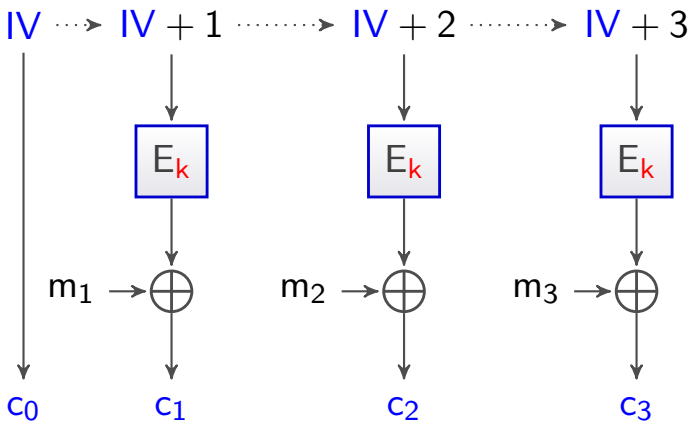
$\Rightarrow$ Some **more** modes-of-operation can be proven secure

Selection of parameters:

- K: An exhaustive search reveals the key in 2^K (offline attacks)
- n: Typically features in bounds related to construction (online attacks)

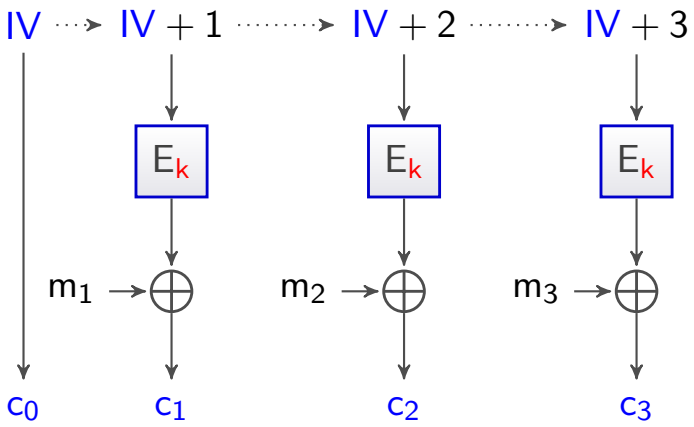
Using (s)PRPs

Decomposing Security



Using (s)PRPs

Decomposing Security



Game (informally):

Adversary picks (m_1, m_2, m_3)

Receives $(c_0, \cdots, c_3)$ which is either true encryption or random.

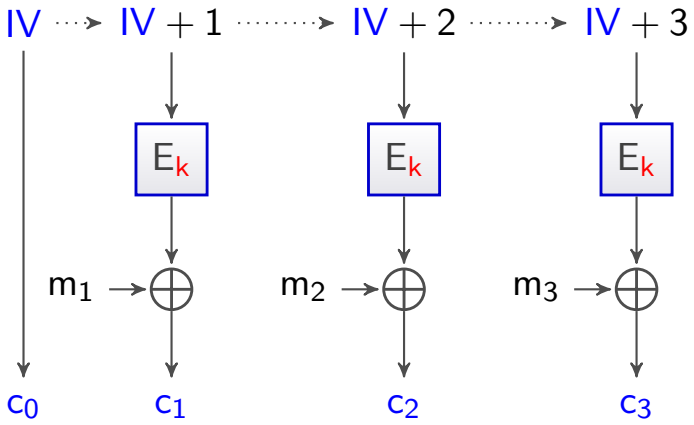
Needs to guess which.

Let's call the advantage

$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A})$$

Using (s)PRPs

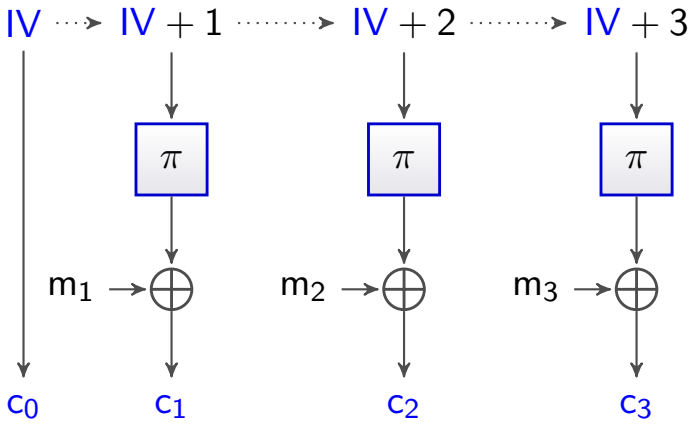
Decomposing Security



Want to analyse the mode independently of the blockcipher!

Using (s)PRPs

Decomposing Security



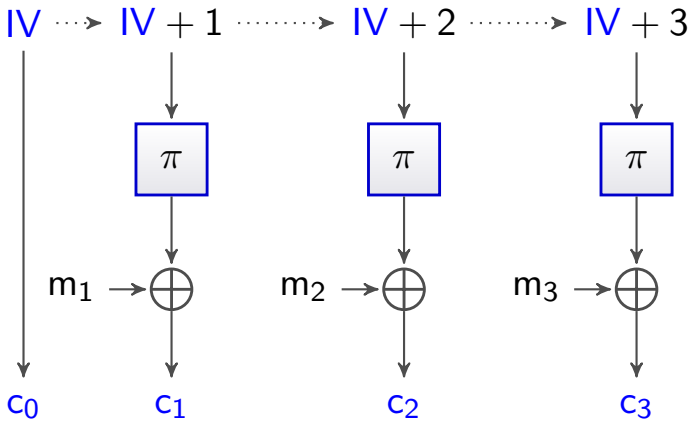
Want to analyse the mode independently of the blockcipher!

Replace the blockcipher by a truly random permutation instead

$$\mathbf{Adv}^{conf}_{CTR[\mathsf{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[\pi]}(\mathbb{A})$$

Using (s)PRPs

Decomposing Security



Want to analyse the mode independently of the blockcipher!

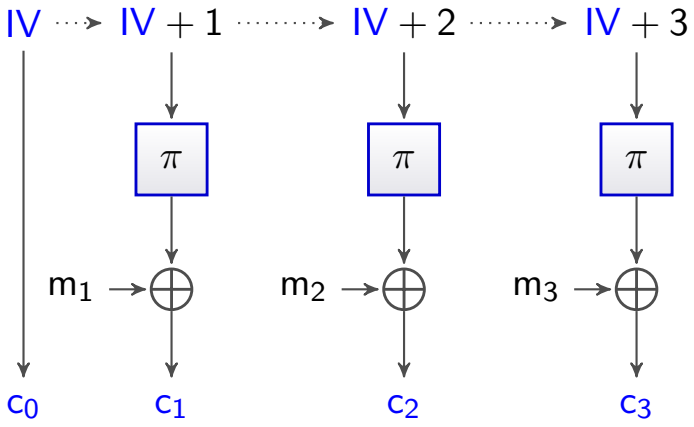
Replace the blockcipher by a truly random permutation instead

If $\mathbb{A}$ could tell the difference, it could win E's prp game.

$$\mathbf{Adv}^{conf}_{CTR[\mathsf{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[\pi]}(\mathbb{A}) + \mathbf{Adv}^{prp}_{\mathsf{E}}(\mathbb{A}')$$

Using (s)PRPs

Decomposing Security



Finally

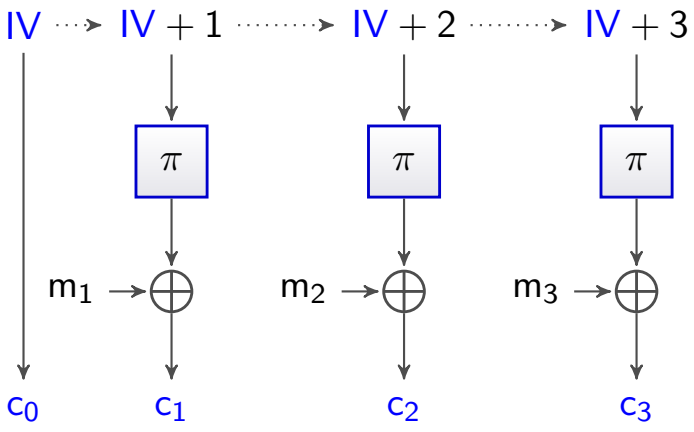
$$\mathbf{Adv}^{conf}_{CTR[\pi]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[\pi]}(1)$$

The latter is the best a computationally unbounded adversary can do with a single query.

$$\mathbf{Adv}^{conf}_{CTR[\mathsf{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[\pi]}(\mathbb{A}) + \mathbf{Adv}^{prp}_{\mathsf{E}}(\mathbb{A}')$$

PRP–PRF Switching

The Lemma



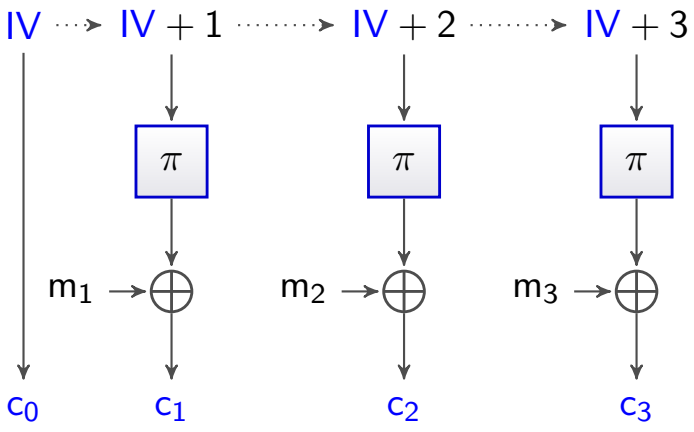
We want to bound

$$\mathbf{Adv}^{conf}_{CTR[\pi]}(1)$$

An information-theoretic problem.

PRP–PRF Switching

The Lemma



We want to bound

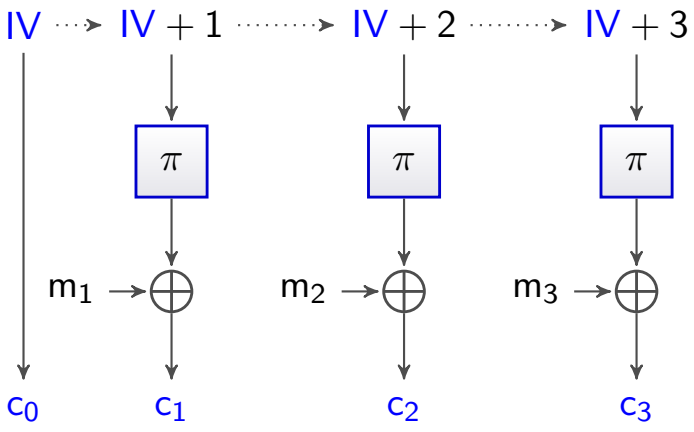
$$\mathbf{Adv}^{conf}_{CTR[\pi]}(1)$$

An information-theoretic problem.

1. The output from $\pi(\mathsf{IV}+1)$ is uniformly random
   $\Rightarrow$ the output c_1 is uniformly random.

PRP–PRF Switching

The Lemma



We want to bound

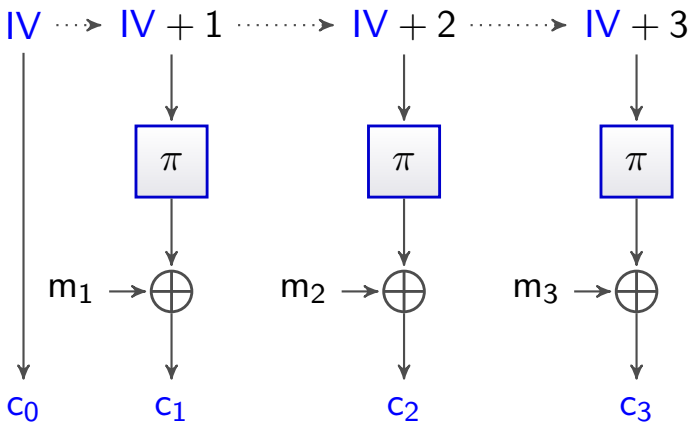
$$\mathbf{Adv}^{conf}_{CTR[\pi]}(1)$$

An information-theoretic problem.

1. The output from $\pi(\mathsf{IV}+1)$ is uniformly random
   $\Rightarrow$ the output c_1 is uniformly random.
2. The output from
   $\pi(\mathsf{IV}+2) \neq \pi(\mathsf{IV}+1)$
   $\Rightarrow$ no longer uniformly random.

PRP–PRF Switching

The Lemma

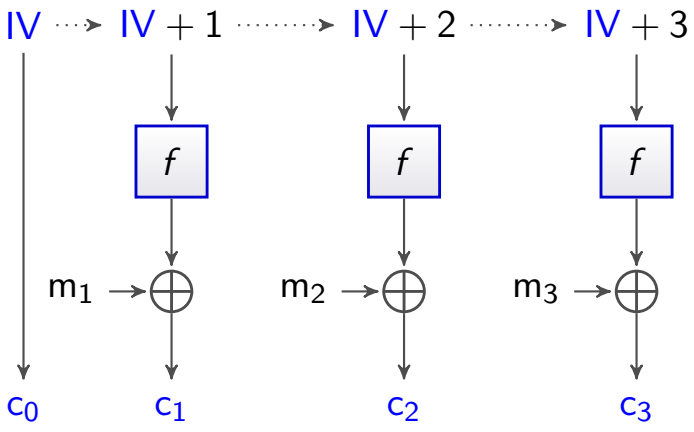


Apart from the first call, the output from any call to π is skewed.

How bad can it get?

PRP–PRF Switching

The Lemma



Apart from the first call, the output from any call to π is skewed.

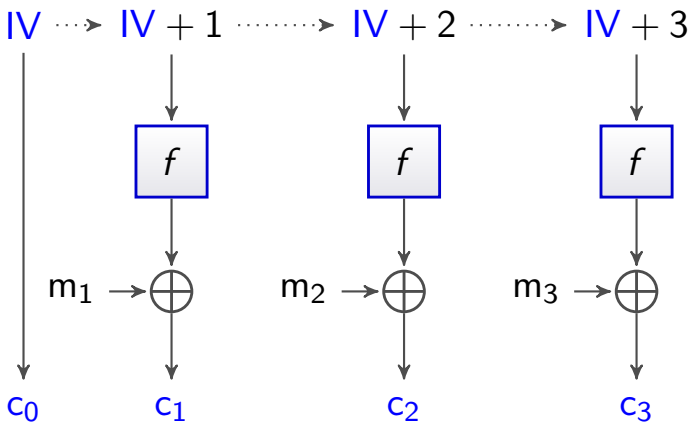
How bad can it get?

Replace the random permutation by a random function instead

$$\mathbf{Adv}^{conf}_{CTR[\pi]}(1) \leq \mathbf{Adv}^{conf}_{CTR[f]}(1)$$

PRP–PRF Switching

The Lemma



Apart from the first call, the output from any call to π is skewed.

How bad can it get?

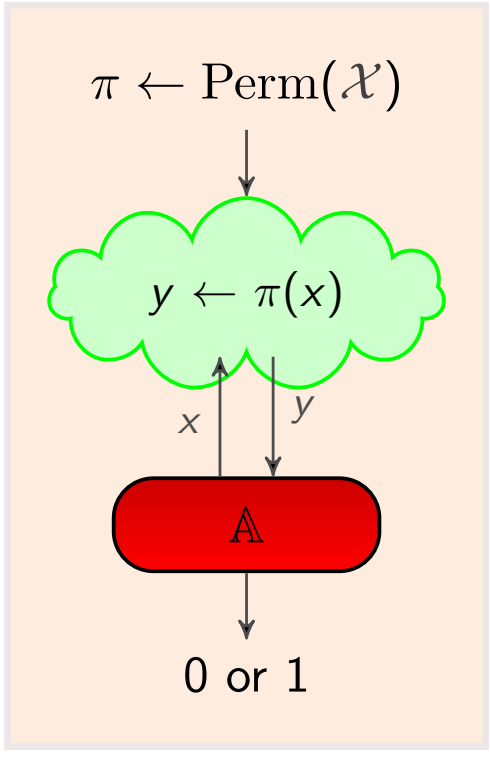
Replace the random permutation by a random function instead

If $\mathbb{A}$ could tell the difference, it would distinguish these using only three queries.

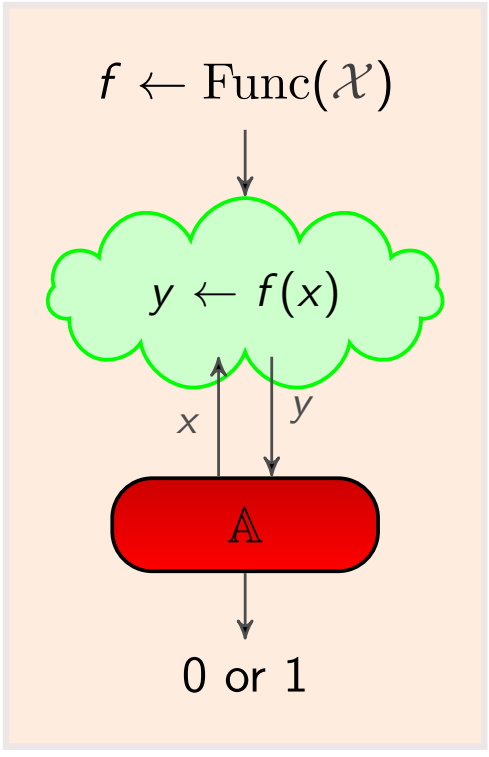
$$\mathbf{Adv}^{conf}_{CTR[\pi]}(1) \leq \mathbf{Adv}^{conf}_{CTR[f]}(1) + \Delta^{f}_{\pi}(3)$$

PRP–PRF Switching

The Bound



v



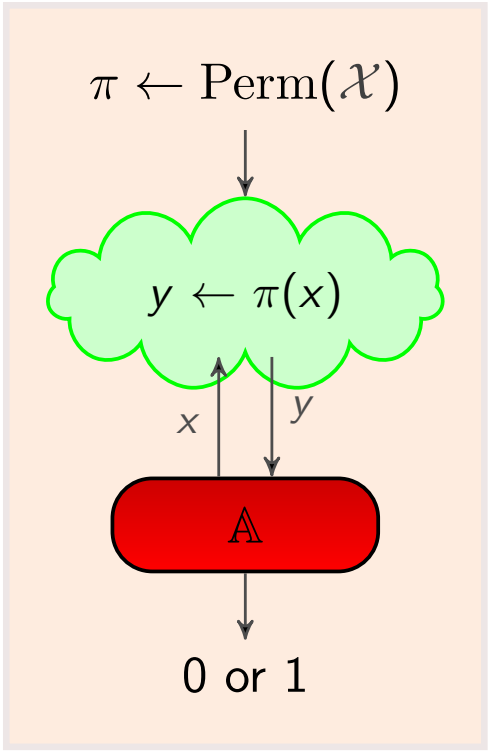
Random permutation world

Random function world

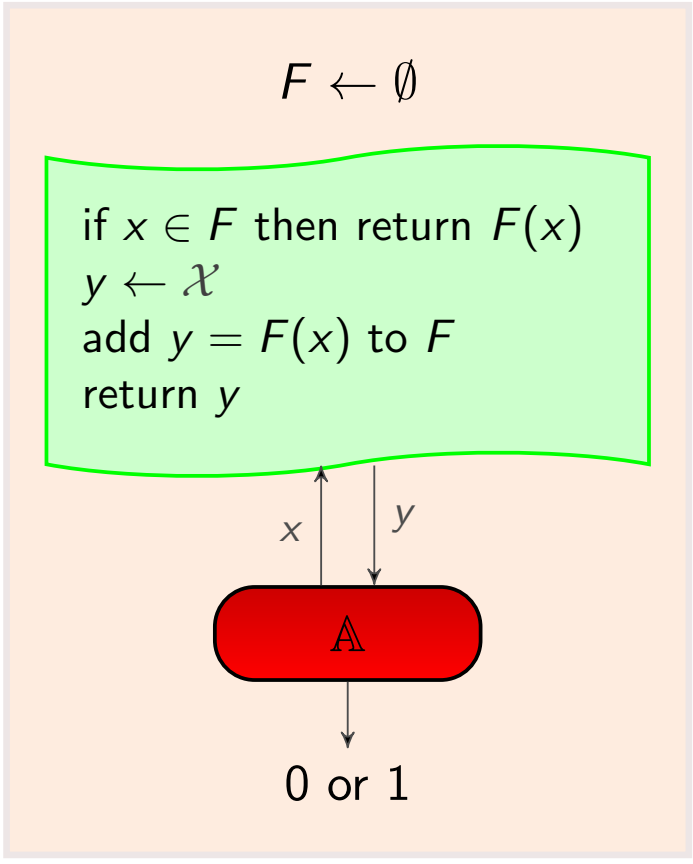
$$\Delta_{\pi}^{f} = \left| \mathbf{Pr}\left[\mathbb{A}^{\pi} = 0\right] - \mathbf{Pr}\left[\mathbb{A}^{f} = 0\right] \right|$$

PRP–PRF Switching

The Bound



Random permutation world

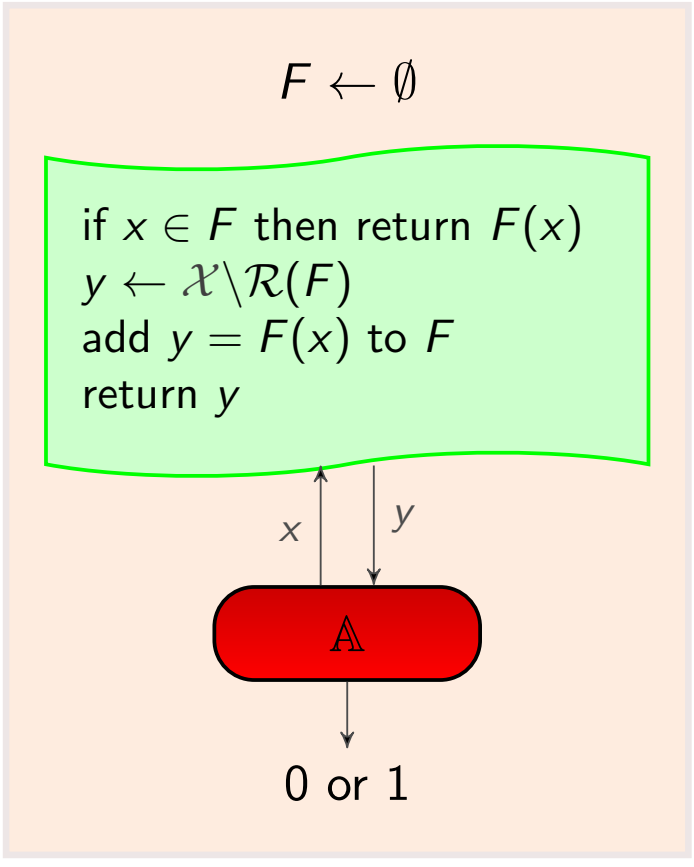


Random function world

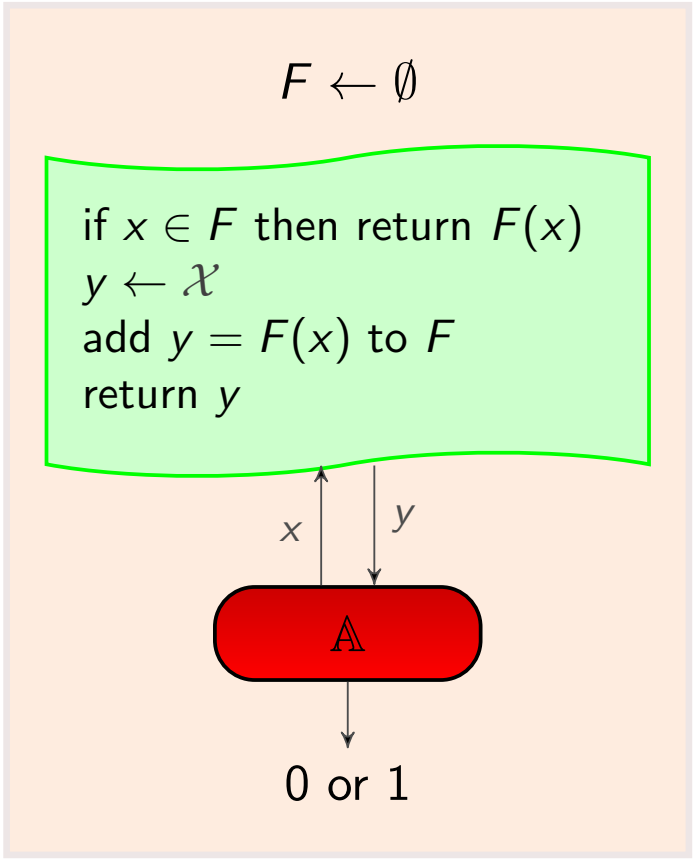
$$\Delta_{\pi}^{f} = \left| \mathbf{Pr}\left[\mathbb{A}^{\pi} = 0\right] - \mathbf{Pr}\left[\mathbb{A}^{f} = 0\right] \right|$$

PRP–PRF Switching

The Bound



Random permutation world

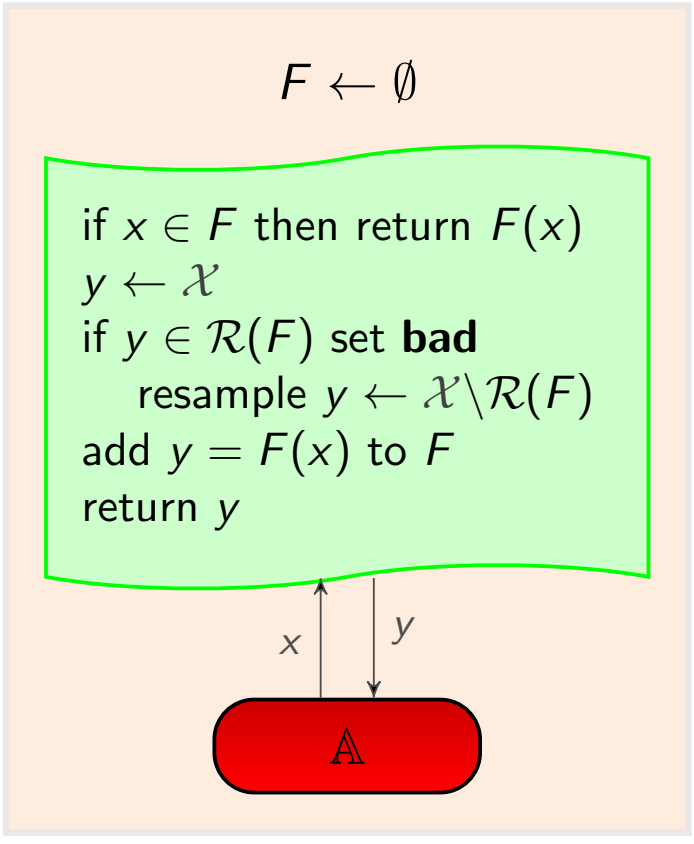


Random function world

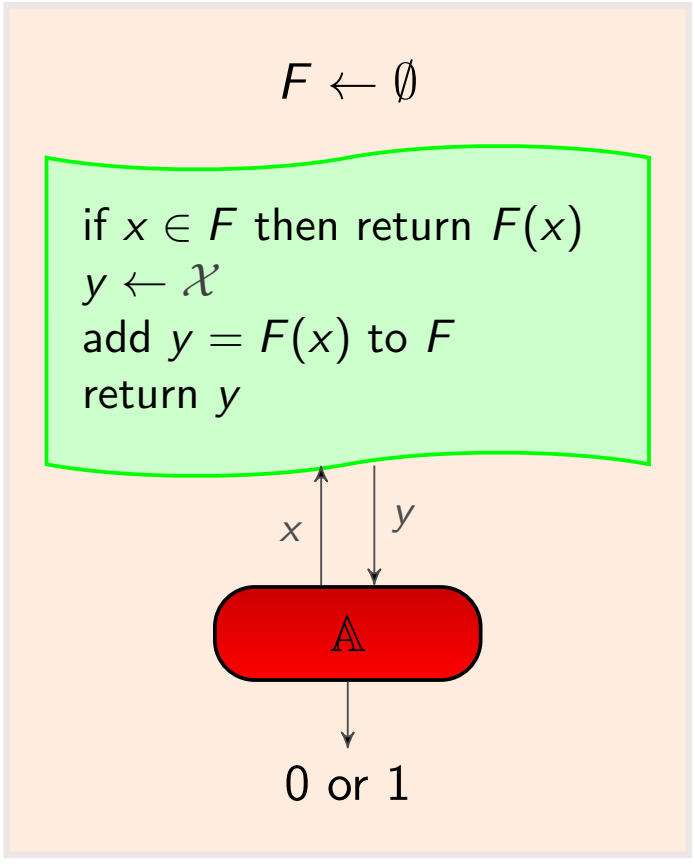
Lazy sampling instead of sampling entire functions

PRP–PRF Switching

The Bound



Random permutation world

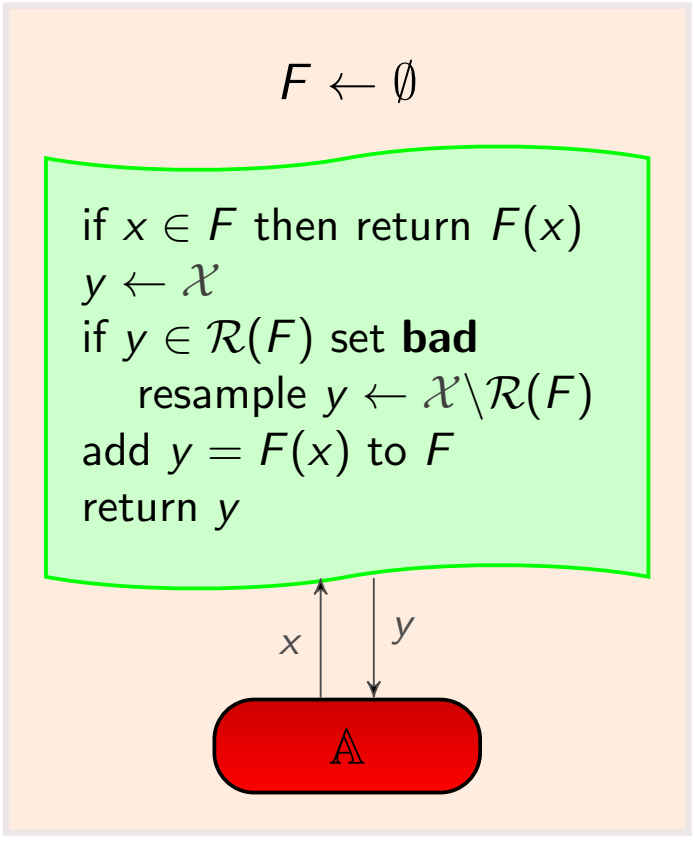


Random function world

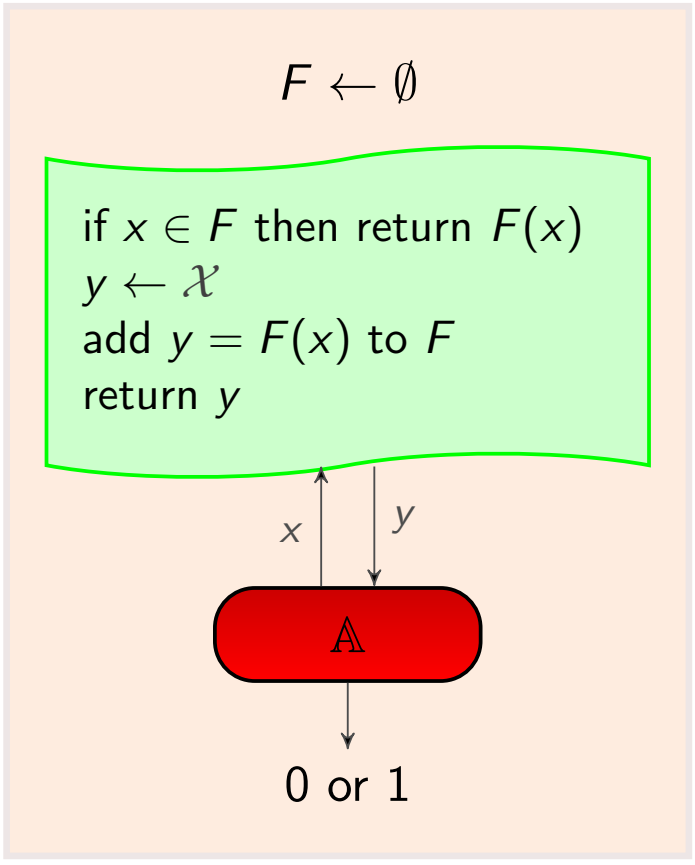
The two worlds are identical-until-**bad**

PRP–PRF Switching

The Bound



Random permutation world

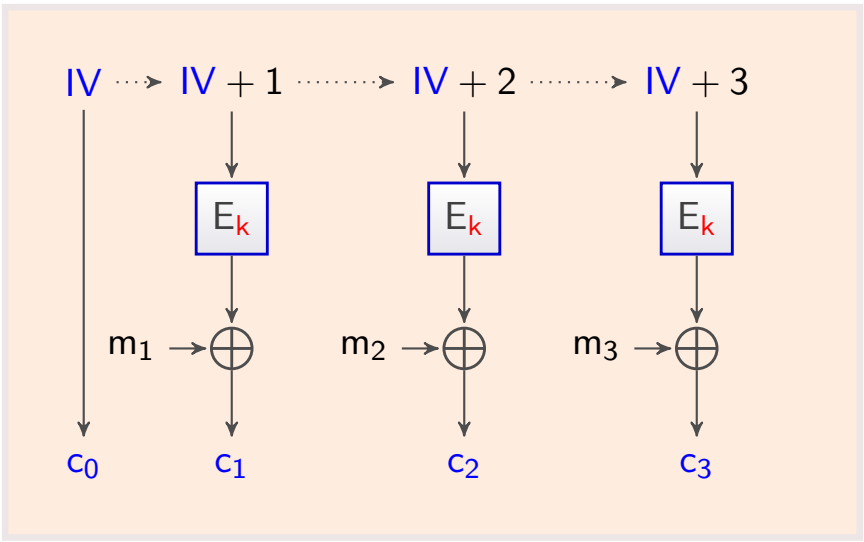


Random function world

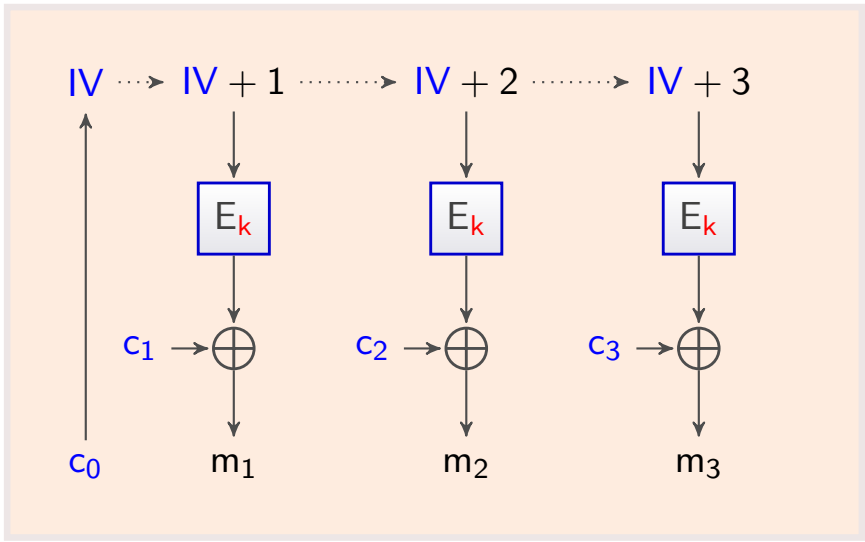
$$\Delta_{\pi}^{f} = \mathbf{Pr}\left[\mathbb{A} \text{ sets } bad\right] \leq q^2/2^{n+1}$$

Using PRPs

Brief Recap



Encryption

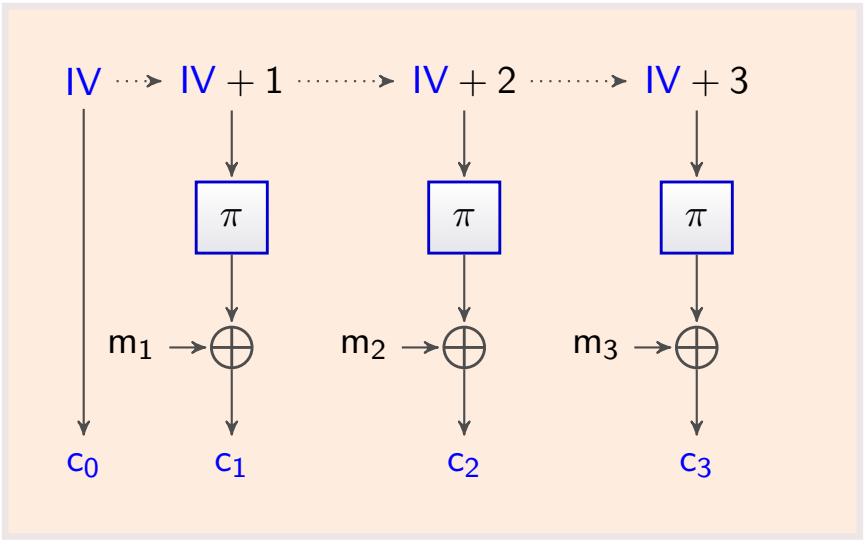


Decryption

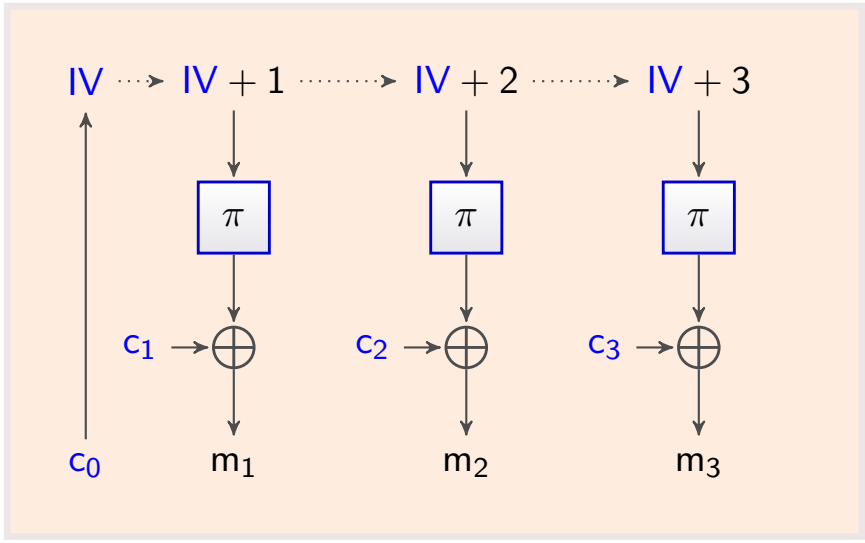
$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A})$$

Using PRPs

Brief Recap



Encryption

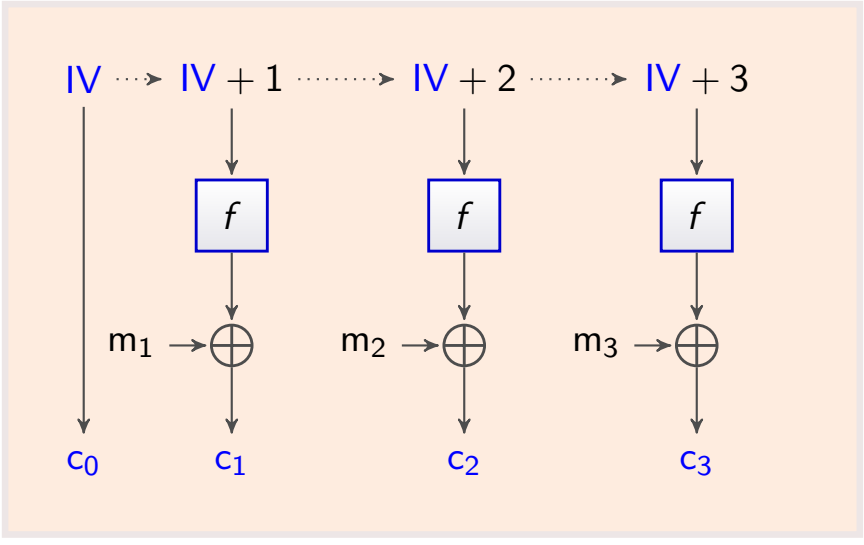


Decryption

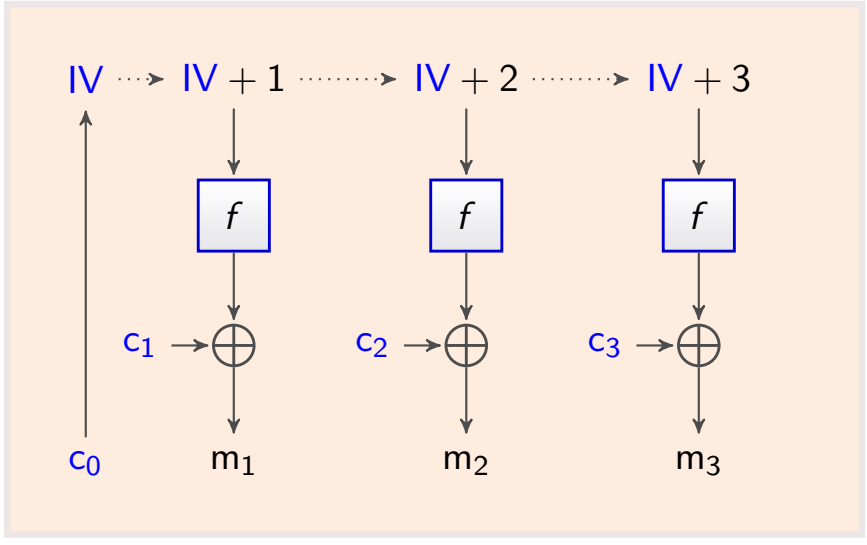
$$\mathbf{Adv}^{conf}_{CTR[\mathsf{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[\pi]}(q_E, q_D) + \mathbf{Adv}^{prp}_{\mathsf{E}}(\mathbb{A}')$$

Using PRPs

Brief Recap



Encryption

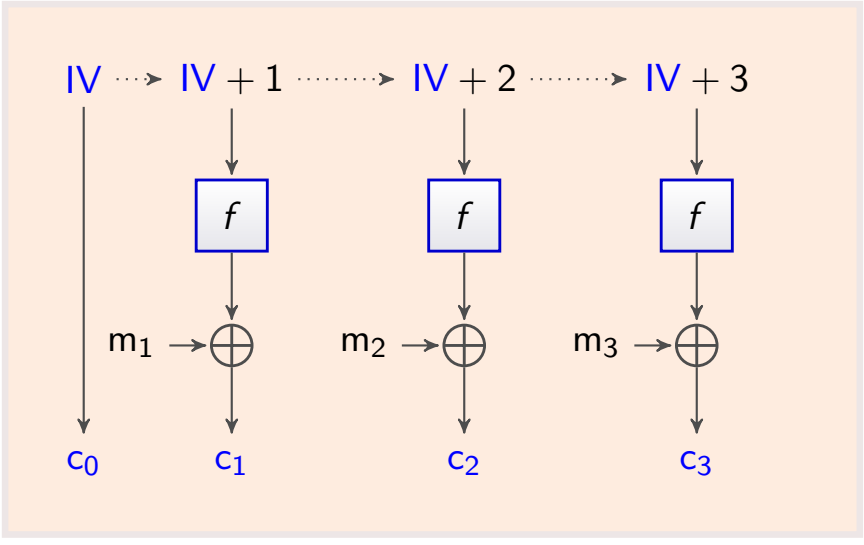


Decryption

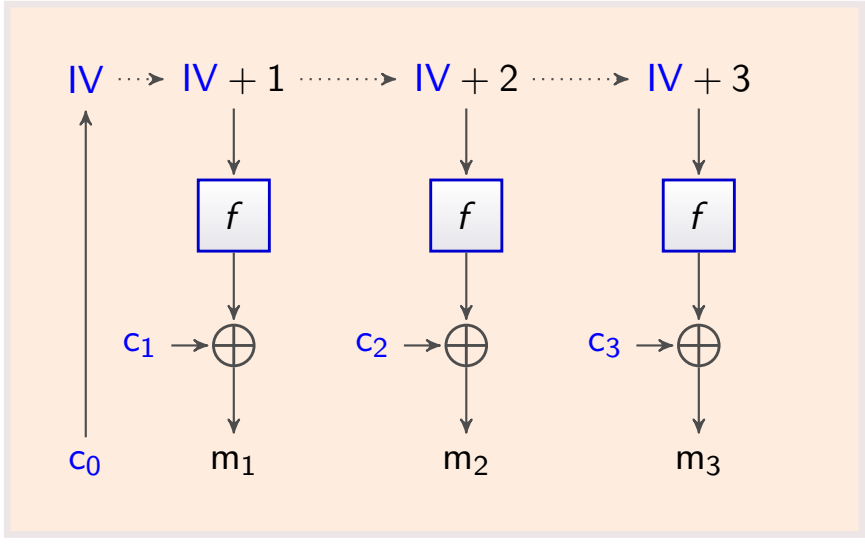
$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[f]}(q_E, q_D) + \Delta^{f}_{\pi}(3q_E + 3q_D) + \mathbf{Adv}^{prp}_{\mathrm{E}}(\mathbb{A}')$$

Using PRPs

Brief Recap



Encryption

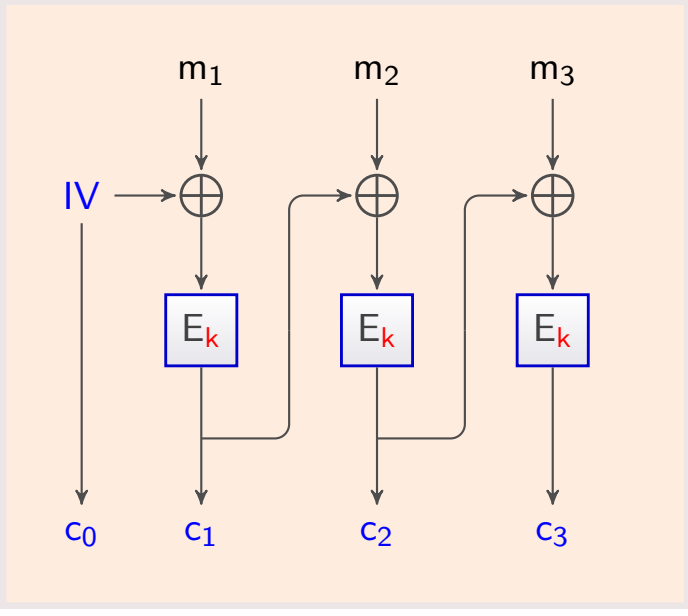


Decryption

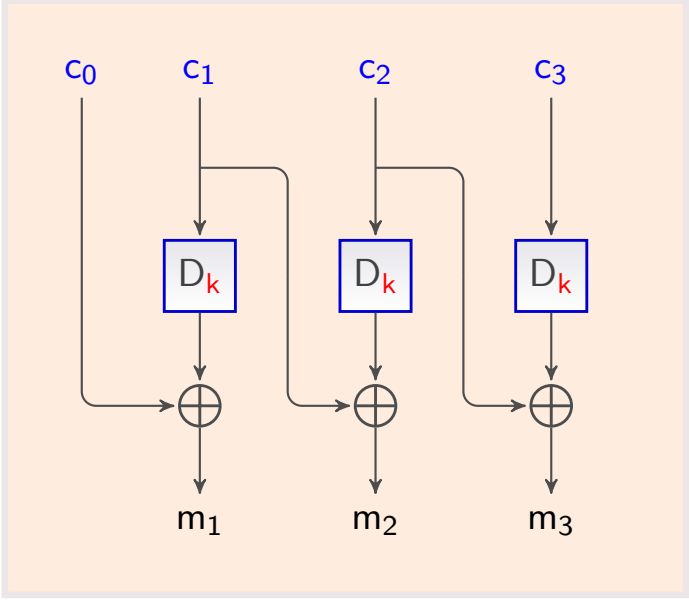
$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[f]}(q_E, q_D) + (3q_E + 3q_D)^2/2^{n+1} + \mathbf{Adv}^{prp}_{\mathrm{E}}(\mathbb{A}')$$

Using strong PRPs

CBC Encryption Case Study



Encryption

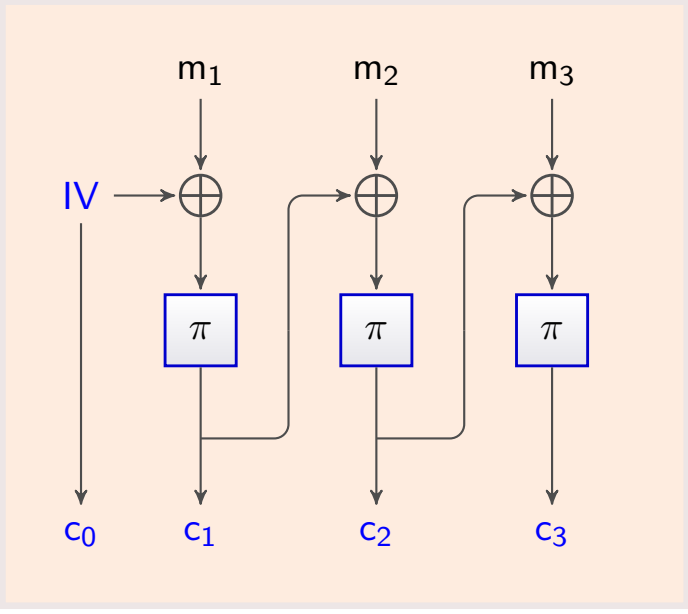


Decryption

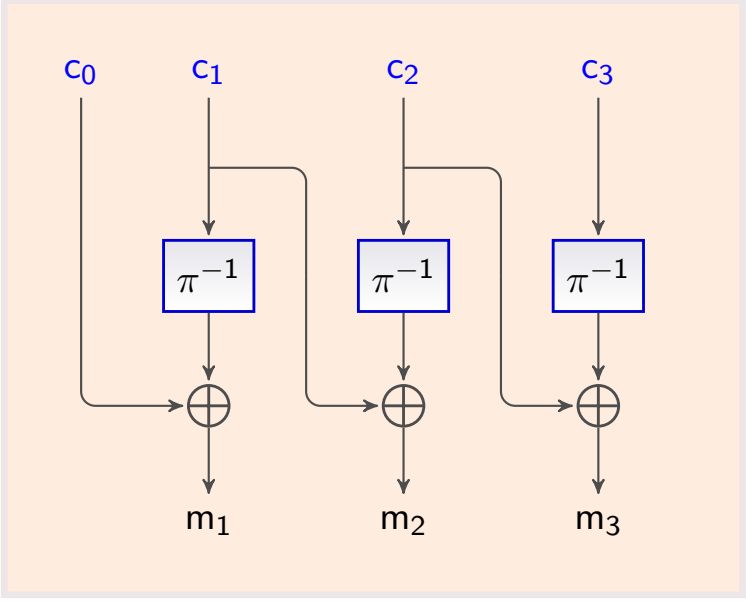
$$\mathbf{Adv}^{conf}_{CBC[E]}(\mathbb{A})$$

Using strong PRPs

CBC Encryption Case Study



Encryption

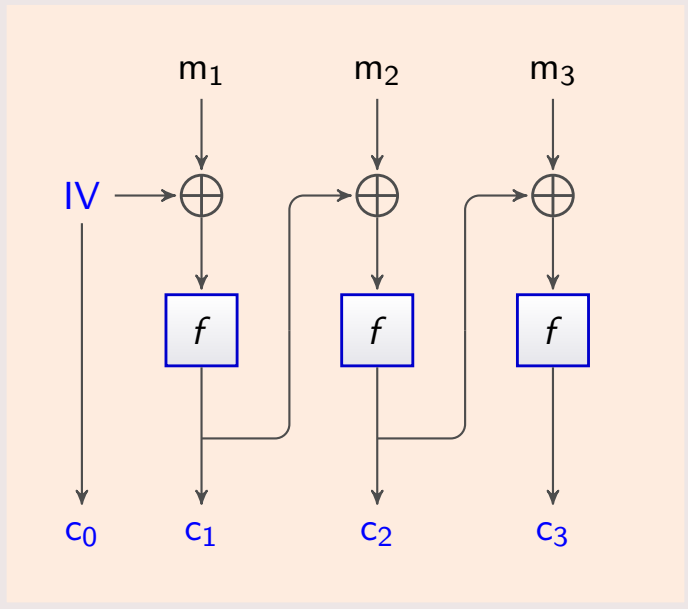


Decryption

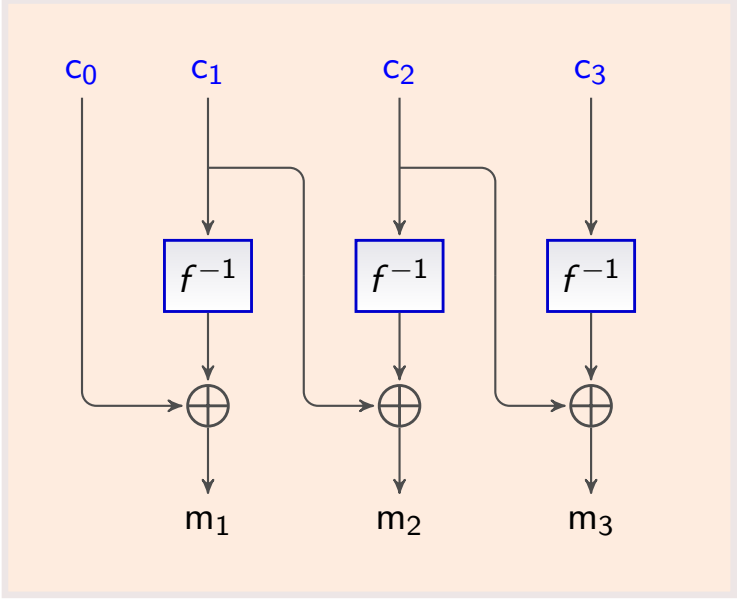
$$\mathbf{Adv}^{conf}_{CBC[\mathsf{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CBC[\pi]}(q_E, q_D) + \mathbf{Adv}^{sprp}_{\mathsf{E}}(\mathbb{A}')$$

Using strong PRPs

CBC Encryption Case Study



Encryption

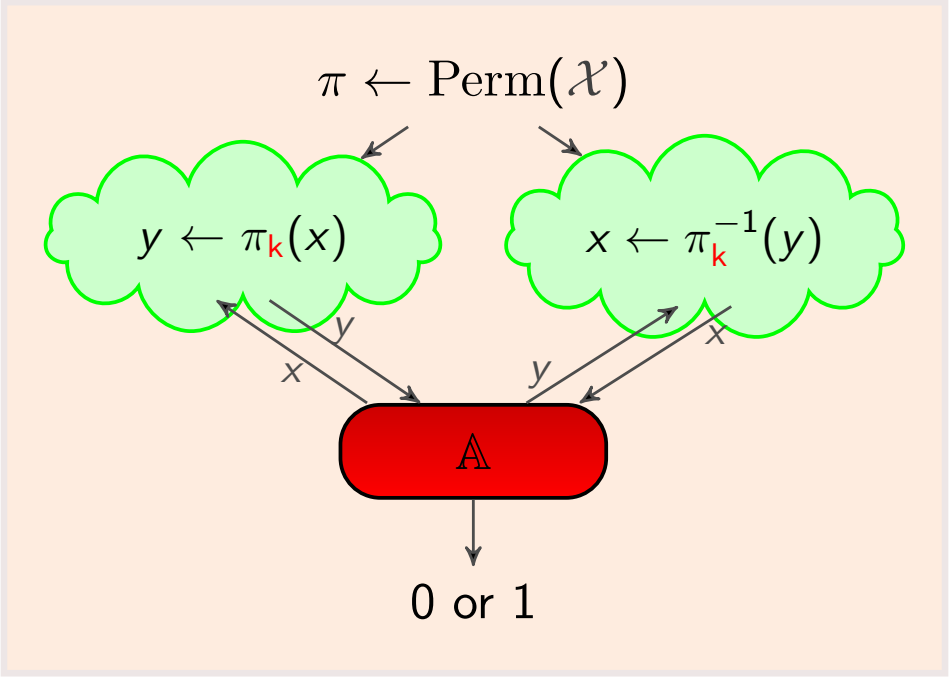


Decryption

$$\mathbf{Adv}^{conf}_{CBC[\mathrm{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CBC[f]}(q_E, q_D) + \Delta^{f,f^{-1}}_{\pi,\pi^{-1}}(3q_E, 3q_D) + \mathbf{Adv}^{sprp}_{\mathrm{E}}(\mathbb{A}')$$

SPRP–SPRF Switching

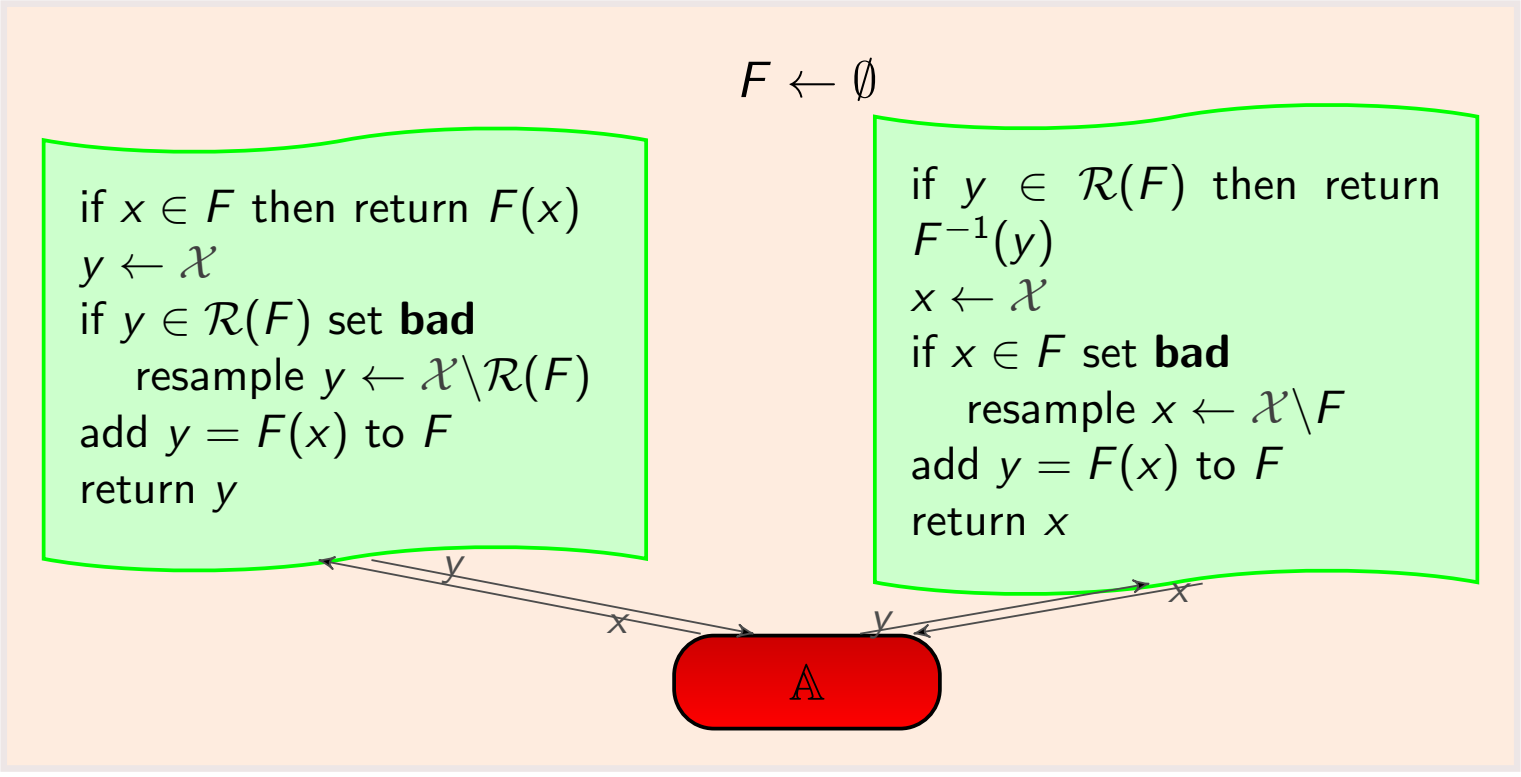
Defining an SPRF



Random permutation world

SPRP–SPRF Switching

Defining an SPRF

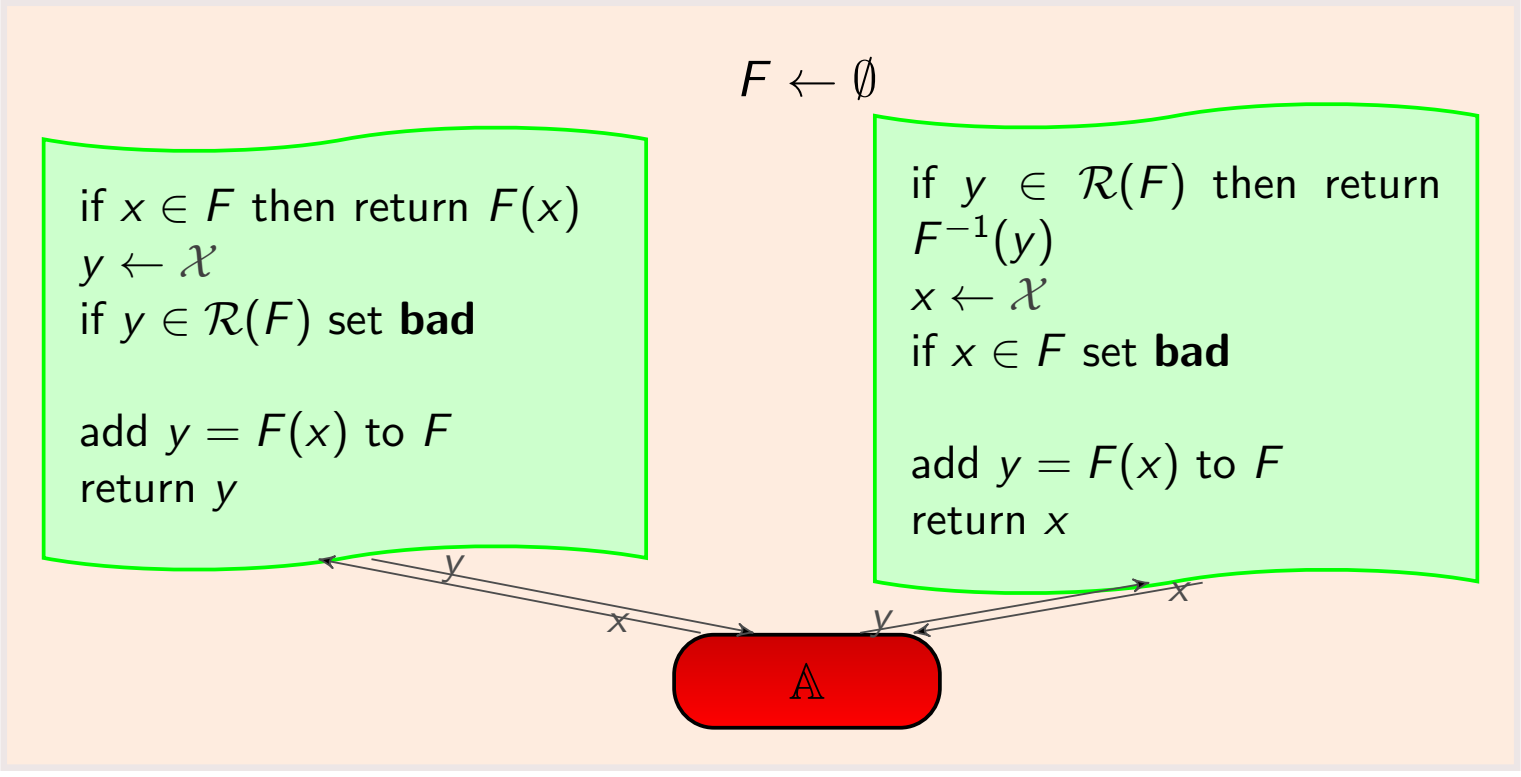


Random permutation world

A random permutation with inverse using lazy sampling

SPRP–SPRF Switching

Defining an SPRF



Random function world

A random function with inverse defined by lazy sampling

Using PRPs

What it all means

We derived the bound

$$\mathbf{Adv}^{conf}_{CTR[\mathsf{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[f]}(q_E, q_D) + (3q_E + 3q_D)^2/2^{n+1} + \mathbf{Adv}^{prp}_{\mathsf{E}}(\mathbb{A}')$$

Advantages

1. Using PRPs as notion allows modular analysis:

Using PRPs

What it all means

We derived the bound

$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[f]}(q_E, q_D) + (3q_E + 3q_D)^2/2^{n+1} + \mathbf{Adv}^{prp}_{\mathrm{E}}(\mathbb{A}')$$

Advantages

1. Using PRPs as notion allows modular analysis:
   - The computational PRP security of the blockcipher

PRP Security depends on K and n;
Offline attacks are relevant.

Using PRPs

What it all means

We derived the bound

$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[f]}(q_E, q_D) + (3q_E + 3q_D)^2/2^{n+1} + \mathbf{Adv}^{prp}_{\mathrm{E}}(\mathbb{A}')$$

Advantages

1. Using PRPs as notion allows modular analysis:
   - The computational PRP security of the blockcipher
   - The information-theoretic security of the construction

No longer depends on K, still on n;
Online attack, number of queries q is important.

Using PRPs

What it all means

We derived the bound

$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[f]}(q_E, q_D) + (3q_E + 3q_D)^2/2^{n+1} + \mathbf{Adv}^{prp}_{\mathrm{E}}(\mathbb{A}')$$

Advantages

1. Using PRPs as notion allows modular analysis:
   - The computational PRP security of the blockcipher
   - The information-theoretic security of the construction
   - A combinatorial birthday-bound
2. The birthday bound implies 64-bit blocks (DES) require care

No longer depends on K, still on n;
Online attack, number of blockcipher calls is important.
Often bounded loosely by $q \cdot L$.

Using PRPs

What it all means

We derived the bound

$$\mathbf{Adv}^{conf}_{CTR[\mathrm{E}]}(\mathbb{A}) \leq \mathbf{Adv}^{conf}_{CTR[f]}(q_E, q_D) + (3q_E + 3q_D)^2/2^{n+1} + \mathbf{Adv}^{prp}_{\mathrm{E}}(\mathbb{A}')$$

Advantages

1. Using PRPs as notion allows modular analysis:
   - The computational PRP security of the blockcipher
   - The information-theoretic security of the construction
   - A combinatorial birthday-bound
2. The birthday bound implies 64-bit blocks (DES) require care
3. SPRP security is needed when inverse blockcipher calls are made

Ciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$$

where

- $\mathcal{K} = \{0,1\}^{\mathsf{K}}$ is the set of keys,
- $\mathcal{X} = \{0,1\}^{\mathsf{n}}$ the set of plaintext blocks

Ciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K} = \{0,1\}^{\mathsf{K}}$ is the set of keys,
- $\mathcal{X} = \{0,1\}^{\mathsf{n}}$ the set of plaintext blocks

Ciphers

For a cipher, $\mathcal{X} = \{0,1\}^*$ instead

1. Still require that $|\mathsf{E}_k(x)| = |x|$
2. Some ciphers only support a subset of lengths
3. For (s)PRP security, adversary is not bound to particular input length.

Tweakable blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K} = \{0,1\}^{\mathsf{K}}$ is the set of keys,
- $\mathcal{X} = \{0,1\}^{\mathsf{n}}$ the set of plaintext blocks

Tweakable blockciphers

Syntax

A blockcipher is a set of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K} = \{0,1\}^{\mathsf{K}}$ is the set of keys,
- $\mathcal{X} = \{0,1\}^{\mathsf{n}}$ the set of plaintext blocks

Tweakable Blockciphers

For a cipher, $\mathsf{E} : \mathcal{K} \times \mathcal{T}\mathcal{X} \to \mathcal{X}$ instead

1. The input $T \in \mathcal{T}$ is called a tweak
2. Now require that $\mathsf{E}_k^T(\cdot)$ a permutation
3. For (s)PRP security, adversary has full control of tweak

Tweakable blockciphers

Advantages

A tweakable blockcipher is a family of keyed permutations

$$\mathsf{E} : \mathcal{K} \times \mathcal{T}\mathcal{X} \to \mathcal{X}$$

where

- $\mathcal{K} = \{0,1\}^{\mathsf{K}}$ is the set of keys,
- $\mathcal{T} \subseteq \{0,1\}^*$ is the set of tweaks,
- $\mathcal{X} = \{0,1\}^{\mathsf{n}}$ the set of plaintext blocks

Main benefits of tweaks

- Efficiency: retweaking is faster than rekeying
- Security: retweaking is cleaner than rekeying
- Tightness: Uniqueness of tweaks means no PRP–PRF switching needed

Advanced Standard Notions

When the key is not secret

ICM Ideal Cipher Model

Give adversary control over key and plaintext;
compare with family of random permutations.

Used to prove heuristic properties of blockcipher-based hashing.

IPM Ideal Permutation Model

Fix a public key, compare with random permutation;

Used to prove heuristic properties of sponge constructions.

KKA Known Key Attacks

AES_0 strictly speaking is not a random permutation (e.g. it has no entropy) but what kind of behaviour would be atypical?

Relevant to compare (theoretical) attacks on AES.

Advanced Standard Notions

When the key is (re)used elsewhere

RKA **Related Key Attacks**
Relevant when for instance using both E_k and $\mathsf{E}_{k \oplus 1}$;
Some protocols might trigger this behaviour.

KDM **Key Dependent Message Attacks**
Relevant when (parts of) the key get encrypted under itself;
can happen in modes-of-operation for disk-encryption.

SCA **Side Channel Attacks**
Implementations might leak partial information about the key,
the goal of SCA is usually key-recovery; effect on PRP strength unclear.