How to Construct an Ideal Cipher from a Small Set of Public Permutations

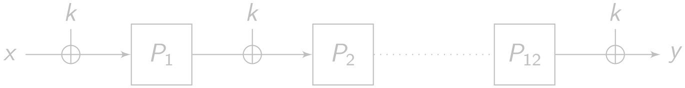
Rodolphe Lampe and Yannick Seurin

University of Versailles and ANSSI

ASIACRYPT 2013 — December 3, 2013

Summary

- We show how to construct an ideal cipher from a small set of n-bit public random permutations $\{P_1, \ldots, P_r\}$
- The construction we consider is the single-key iterated Even-Mansour cipher (*aka* key-alternating cipher) with 12 rounds:

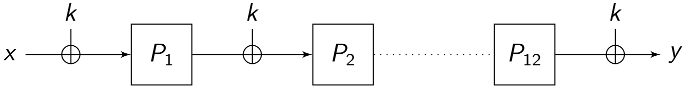


$\Rightarrow$ this yields a family of 2^n permutations indexed by the n-bit key k from only 12 public n-bit permutations

- We show that this construction "behaves" as an ideal cipher with n-bit blocks and n-bit keys using the indifferentiability framework
- We also show that at least 4 rounds are necessary to achieve indifferentiability from an ideal cipher

Summary

- We show how to construct an ideal cipher from a small set of n-bit public random permutations $\{P_1, \ldots, P_r\}$
- The construction we consider is the single-key iterated Even-Mansour cipher (*aka* key-alternating cipher) with 12 rounds:

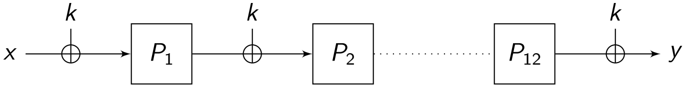


$\Rightarrow$ this yields a family of 2^n permutations indexed by the n-bit key k from only 12 public n-bit permutations

- We show that this construction "behaves" as an ideal cipher with n-bit blocks and n-bit keys using the indifferentiability framework
- We also show that at least 4 rounds are necessary to achieve indifferentiability from an ideal cipher

Summary

- We show how to construct an ideal cipher from a small set of n-bit public random permutations $\{P_1, \ldots, P_r\}$
- The construction we consider is the single-key iterated Even-Mansour cipher (*aka* key-alternating cipher) with 12 rounds:

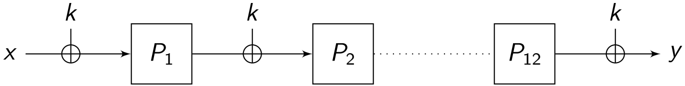


$\Rightarrow$ this yields a family of 2^n permutations indexed by the n-bit key k from only 12 public n-bit permutations

- We show that this construction "behaves" as an ideal cipher with n-bit blocks and n-bit keys using the indifferentiability framework
- We also show that at least 4 rounds are necessary to achieve indifferentiability from an ideal cipher

Summary

- We show how to construct an ideal cipher from a small set of n-bit public random permutations $\{P_1, \ldots, P_r\}$
- The construction we consider is the single-key iterated Even-Mansour cipher (*aka* key-alternating cipher) with 12 rounds:



$\Rightarrow$ this yields a family of 2^n permutations indexed by the n-bit key k from only 12 public n-bit permutations

- We show that this construction "behaves" as an ideal cipher with n-bit blocks and n-bit keys using the indifferentiability framework
- We also show that at least 4 rounds are necessary to achieve indifferentiability from an ideal cipher

Outline

1. Background on the Iterated Even-Mansour Cipher
2. Indifferentiability of the IEM cipher
   - Formalizing the problem
   - Which key schedule?
   - At least 4 rounds are necessary
3. Indifferentiability proof for 12 rounds

Outline

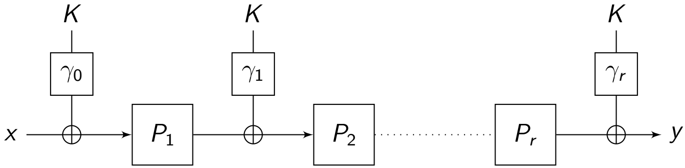
1. Background on the Iterated Even-Mansour Cipher

2. Indifferentiability of the IEM cipher
   - Formalizing the problem
   - Which key schedule?
   - At least 4 rounds are necessary

3. Indifferentiability proof for 12 rounds

Iterated Even-Mansour cipher (*aka* key-alternating cipher)

Iterated Even-Mansour (IEM) with r rounds:

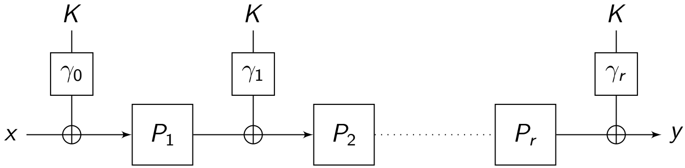


- The P_i's are public permutations on $\{0,1\}^n$
- $K \in \{0,1\}^\ell$ is the (master) key
- The γ_i's are key derivation functions mapping K to n-bit values

Also named key-alternating cipher

Iterated Even-Mansour cipher (*aka* key-alternating cipher)

Iterated Even-Mansour (IEM) with r rounds:



- The P_i's are public permutations on $\{0,1\}^n$
- $K \in \{0,1\}^\ell$ is the (master) key
- The γ_i's are key derivation functions mapping K to n-bit values

Also named key-alternating cipher

Iterated Even-Mansour cipher (*aka* key-alternating cipher)

Most (if not all) SPN ciphers can be described as key-alternating ciphers. E.g. for AES-128, one has $r = 10$, the γ_i's are efficiently invertible permutations, and:

$$P_1 = \ldots = P_9 = \mathtt{SubBytes} \circ \mathtt{ShiftRows} \circ \mathtt{MixColumns}$$
$$P_{10} = \mathtt{SubBytes} \circ \mathtt{ShiftRows}$$

When the P_i's are fixed permutations, one can prove results like:

- the best differential characteristic over $r' < r$ rounds has probability at most p
- the best linear approximation over $r' < r$ rounds has probability at most p'

This gives upper bounds on the distinguishing probability of very specific adversaries

Iterated Even-Mansour cipher (*aka* key-alternating cipher)

Most (if not all) SPN ciphers can be described as key-alternating ciphers. E.g. for AES-128, one has $r = 10$, the γ_i's are efficiently invertible permutations, and:

$$P_1 = \ldots = P_9 = \mathtt{SubBytes} \circ \mathtt{ShiftRows} \circ \mathtt{MixColumns}$$
$$P_{10} = \mathtt{SubBytes} \circ \mathtt{ShiftRows}$$

When the P_i's are fixed permutations, one can prove results like:

- the best differential characteristic over $r' < r$ rounds has probability at most p
- the best linear approximation over $r' < r$ rounds has probability at most p'

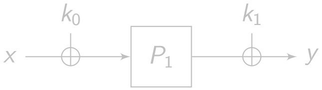
This gives upper bounds on the distinguishing probability of very specific adversaries

Analysis in the Random Permutation Model (RPM)

Recently, a lot of results have been obtained in the Random Permutation Model: the P_i's are viewed as oracles to which the adversary can make black-box queries (both to P_i and P_i^{-1}).

Interpretation: gives a guarantee against any adversary which does not use particular properties of the P_i's

In fact, this model was already considered 15 years ago by Even and Mansour for $r = 1$ round: they showed that the following cipher is pseudorandom up to $\mathcal{O}(2^{n/2})$ queries of the adversary, when P_1 is a public random permutation:

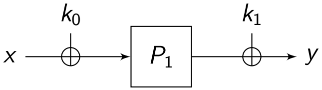


Analysis in the Random Permutation Model (RPM)

Recently, a lot of results have been obtained in the Random Permutation Model: the P_i's are viewed as oracles to which the adversary can make black-box queries (both to P_i and P_i^{-1}).

Interpretation: gives a guarantee against any adversary which does not use particular properties of the P_i's

In fact, this model was already considered 15 years ago by Even and Mansour for $r = 1$ round: they showed that the following cipher is pseudorandom up to $\mathcal{O}(2^{n/2})$ queries of the adversary, when P_1 is a public random permutation:

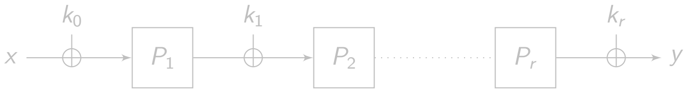


Pseudorandomness of the IEM cipher (in the RPM)

The following results have been successively obtained for the pseudorandomness of the IEM cipher (notation: $N = 2^n$):

- for $r = 1$ round, security up to $\mathcal{O}(N^{\frac{1}{2}})$ queries [EM97]
- for $r \geq 2$, security up to $\mathcal{O}(N^{\frac{2}{3}})$ queries [BKL$^+$12]
- for $r \geq 3$, security up to $\mathcal{O}(N^{\frac{3}{4}})$ queries [Ste13]
- for any even r, security up to $\mathcal{O}(N^{\frac{r}{r+2}})$ queries [LPS12]
- tight result: for r rounds, security up to $\mathcal{O}(N^{\frac{r}{r+1}})$ queries [CS13]

Results for independent round keys $(k_0, k_1, \ldots, k_r)$

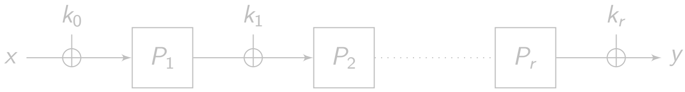


Pseudorandomness of the IEM cipher (in the RPM)

The following results have been successively obtained for the pseudorandomness of the IEM cipher (notation: $N = 2^n$):

- for $r = 1$ round, security up to $\mathcal{O}(N^{\frac{1}{2}})$ queries [EM97]
- for $r \geq 2$, security up to $\mathcal{O}(N^{\frac{2}{3}})$ queries [BKL+12]
- for $r \geq 3$, security up to $\mathcal{O}(N^{\frac{3}{4}})$ queries [Ste13]
- for any even r, security up to $\mathcal{O}(N^{\frac{r}{r+2}})$ queries [LPS12]
- tight result: for r rounds, security up to $\mathcal{O}(N^{\frac{r}{r+1}})$ queries [CS13]

Results for independent round keys $(k_0, k_1, \ldots, k_r)$

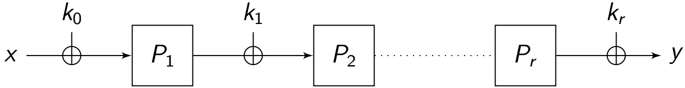


Pseudorandomness of the IEM cipher (in the RPM)

The following results have been successively obtained for the pseudorandomness of the IEM cipher (notation: $N = 2^n$):

- for $r = 1$ round, security up to $\mathcal{O}(N^{\frac{1}{2}})$ queries [EM97]
- for $r \geq 2$, security up to $\mathcal{O}(N^{\frac{2}{3}})$ queries [BKL+12]
- for $r \geq 3$, security up to $\mathcal{O}(N^{\frac{3}{4}})$ queries [Ste13]
- for any even r, security up to $\mathcal{O}(N^{\frac{r}{r+2}})$ queries [LPS12]
- tight result: for r rounds, security up to $\mathcal{O}(N^{\frac{r}{r+1}})$ queries [CS13]

Results for independent round keys $(k_0, k_1, \ldots, k_r)$



Outline

Outline

From indistinguishability to indifferentiability

Previous results state that the IEM cipher is a (strong) pseudorandom permutation (in the random permutation model)
= usual single, secret-key security model

Question

What about related-, known- or chosen-key attacks?
Can we even hope to prove that the IEM "behaves" as (*is indifferentiable from*) an ideal cipher?

Ideal cipher: an independent random permutation for each key

From indistinguishability to indifferentiability

Previous results state that the IEM cipher is a (strong) pseudorandom permutation (in the random permutation model)
= usual single, secret-key security model

Question

What about related-, known- or chosen-key attacks?
Can we even hope to prove that the IEM "behaves" as (*is indifferentiable from*) an ideal cipher?

Ideal cipher: an independent random permutation for each key

From indistinguishability to indifferentiability

Previous results state that the IEM cipher is a (strong) pseudorandom permutation (in the random permutation model)
= usual single, secret-key security model

Question

What about related-, known- or chosen-key attacks?
Can we even hope to prove that the IEM "behaves" as (*is indifferentiable from*) an ideal cipher?

Ideal cipher: an independent random permutation for each key

A word on the ideal cipher model

- the pseudorandomness security notion for a block cipher is sufficient to prove the security of a lot of applications (encryption modes and MACs)
- however, sometimes it is not sufficient (e.g. for block cipher-based hash functions like Davies-Meyer mode)
- ideally, one expects that a good block cipher "behaves" as an independent random permutation for each key
  $\rightarrow$ ideal cipher model: draw an independent perfectly random permutation for each key

A word on the ideal cipher model

- the pseudorandomness security notion for a block cipher is sufficient to prove the security of a lot of applications (encryption modes and MACs)
- however, sometimes it is not sufficient (e.g. for block cipher-based hash functions like Davies-Meyer mode)
- ideally, one expects that a good block cipher "behaves" as an independent random permutation for each key
  $\rightarrow$ ideal cipher model: draw an independent perfectly random permutation for each key

A word on the ideal cipher model

- the pseudorandomness security notion for a block cipher is sufficient to prove the security of a lot of applications (encryption modes and MACs)
- however, sometimes it is not sufficient (e.g. for block cipher-based hash functions like Davies-Meyer mode)
- ideally, one expects that a good block cipher “behaves” as an independent random permutation for each key
  $\rightarrow$ ideal cipher model: draw an independent perfectly random permutation for each key

A word on the ideal cipher model

- similar to the random oracle model for a hash function
- warning: instantiation problems as well (no concrete block cipher can be proved to be an ideal cipher in any reasonable sense)
- though we cannot prove that a block cipher behaves as an ideal cipher in the standard model, we can prove results in idealized models (e.g. the Random Permutation Model in the case of the IEM cipher)
  $\rightarrow$ indifferentiability notion [MRH04]

A word on the ideal cipher model

- similar to the random oracle model for a hash function
- warning: instantiation problems as well (no concrete block cipher can be proved to be an ideal cipher in any reasonable sense)
- though we cannot prove that a block cipher behaves as an ideal cipher in the standard model, we can prove results in idealized models (e.g. the Random Permutation Model in the case of the IEM cipher)
  $\rightarrow$ indifferentiability notion [MRH04]

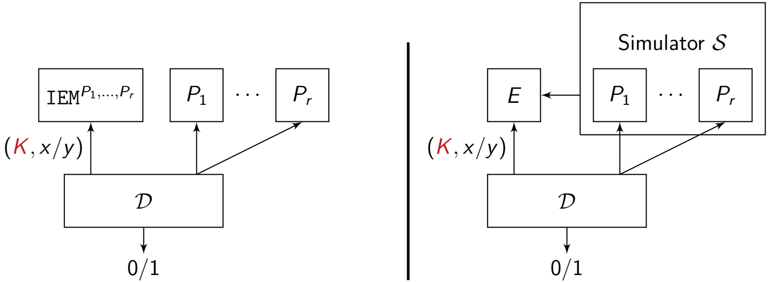
A word on the ideal cipher model

- similar to the random oracle model for a hash function
- warning: instantiation problems as well (no concrete block cipher can be proved to be an ideal cipher in any reasonable sense)
- though we cannot prove that a block cipher behaves as an ideal cipher in the standard model, we can prove results in idealized models (e.g. the Random Permutation Model in the case of the IEM cipher)
  $\rightarrow$ indifferentiability notion [MRH04]

Indifferentiability: definition

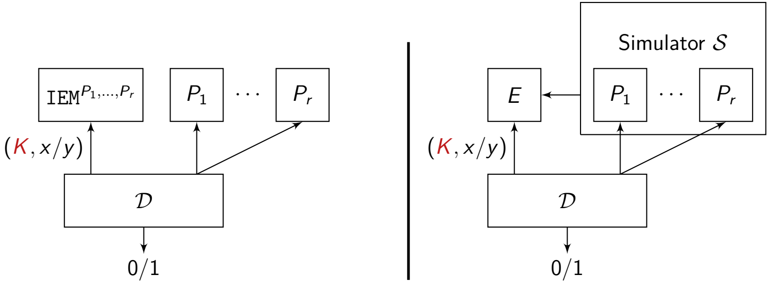
Definition

The IEM cipher $\mathtt{IEM}^{P_1,\ldots,P_r}$ with random permutations $\boldsymbol{P} = (P_1, \ldots, P_r)$ is said indifferentiable from an ideal cipher E if there exists a polynomial time simulator $\mathcal{S}$ with oracle access to E such that the two systems $(\mathtt{IEM}^{\boldsymbol{P}}, \boldsymbol{P})$ and $(E, \mathcal{S}^E)$ are indistinguishable.



Indifferentiability: definition

NB: The distinguisher specifies the plaintext/ciphertext and the key when querying $\mathtt{IEM}^{P_1,\ldots,P_r}$ or E.



The answers of the simulator $\mathcal{S}$ must be:

- coherent with answers the distinguisher can obtain directly from E
- close in distribution to the answers of random permutations

Composition theorem

Usefulness of indifferentiability: composition theorem

Theorem

If a cryptosystem Γ is secure when used with an ideal cipher E, and if $\mathtt{IEM}^{P_1,\ldots,P_r}$ (for sufficiently many rounds) is indifferentiable from E, then Γ is also secure when used with $\mathtt{IEM}^{P_1,\ldots,P_r}$ with random permutations $P_1, \ldots, P_r$ (for single-stage security notions).

Main question

Is the Iterated Even-Mansour cipher, for sufficiently many rounds, and with an adequate key schedule, indifferentiable from an ideal cipher?

Composition theorem

Usefulness of indifferentiability: composition theorem

Theorem

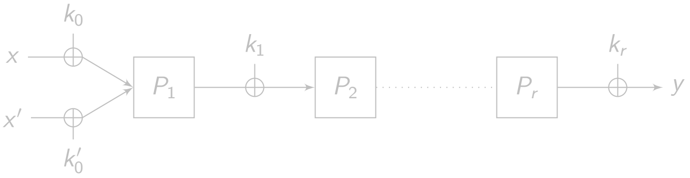
If a cryptosystem Γ is secure when used with an ideal cipher E, and if $\mathtt{IEM}^{P_1,\ldots,P_r}$ (for sufficiently many rounds) is indifferentiable from E, then Γ is also secure when used with $\mathtt{IEM}^{P_1,\ldots,P_r}$ with random permutations $P_1,\ldots,P_r$ (for single-stage security notions).

Main question

Is the Iterated Even-Mansour cipher, for sufficiently many rounds, and with an adequate key schedule, indifferentiable from an ideal cipher?

Outline

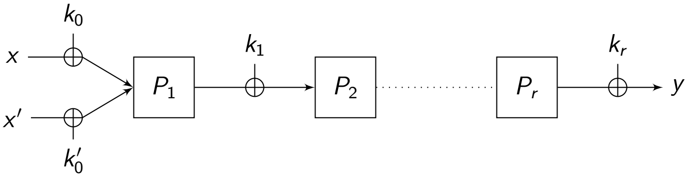
Independent round keys fails(!)



IEM with independent round keys is not indifferentiable from an ideal cipher with key space $\{0,1\}^{(r+1)n}$ because of the following distinguisher:

- choose an arbitrary $x \in \{0,1\}^n$ and $k_0 \in \{0,1\}^n$
- define $x' = x \oplus c$ and $k'_0 = k_0 \oplus c$ with c a non-zero constant
- let $K = (k_0, k_1, \ldots, k_r)$ and $K' = (k'_0, k_1, \ldots, k_r)$
- then $\mathtt{IEM}(K, x) = \mathtt{IEM}(K', x')$
- this holds only with negligible probability for an ideal cipher

Independent round keys fails(!)



IEM with independent round keys is not indifferentiable from an ideal cipher with key space $\{0,1\}^{(r+1)n}$ because of the following distinguisher:

- choose an arbitrary $x \in \{0,1\}^n$ and $k_0 \in \{0,1\}^n$
- define $x' = x \oplus c$ and $k'_0 = k_0 \oplus c$ with c a non-zero constant
- let $K = (k_0, k_1, \ldots, k_r)$ and $K' = (k'_0, k_1, \ldots, k_r)$
- then $\mathtt{IEM}(K, x) = \mathtt{IEM}(K', x')$
- this holds only with negligible probability for an ideal cipher

Proving indifferentiability for the IEM cipher

Independent keys leave too much “freedom” to the adversary.

Two ideas to solve the problem:

1. add a key schedule, and put some cryptographic assumption on it
   $\Rightarrow$ Andreeva et al. CRYPTO 2013 [ABD$^+$13]
2. restrain the key space and correlate the round keys, e.g. $(k, k, \ldots, k)$
   $\Rightarrow$ this paper

Proving indifferentiability for the IEM cipher

Independent keys leave too much “freedom” to the adversary.

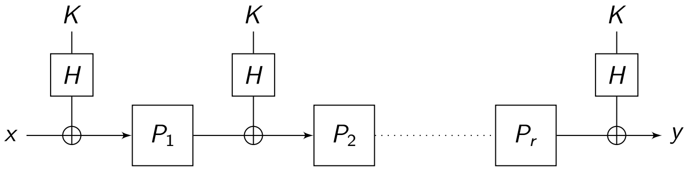
Two ideas to solve the problem:

1. add a key schedule, and put some cryptographic assumption on it
   $\Rightarrow$ Andreeva et al. CRYPTO 2013 [ABD+13]

2. restrain the key space and correlate the round keys, e.g. $(k, k, \ldots, k)$
   $\Rightarrow$ this paper

The [ABD$^+$13] result

IEM with a key-derivation function modeled as a random oracle from $\{0,1\}^\ell$ to $\{0,1\}^n$ (that the adversary queries in a black-box way)



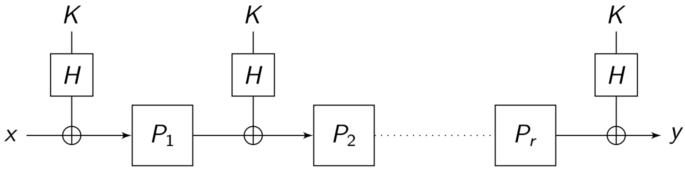
$\rightarrow$ indifferentiable from an ideal cipher with ℓ-bit keys for $r = 5$ ([ABD$^+$13] gives attacks up to 3 rounds)

Better bounds and less rounds than in this paper.

But the assumption about the key derivation is very strong and far from concrete designs (the key-schedule is often invertible)

The [ABD$^+$13] result

IEM with a key-derivation function modeled as a random oracle from $\{0,1\}^\ell$ to $\{0,1\}^n$ (that the adversary queries in a black-box way)



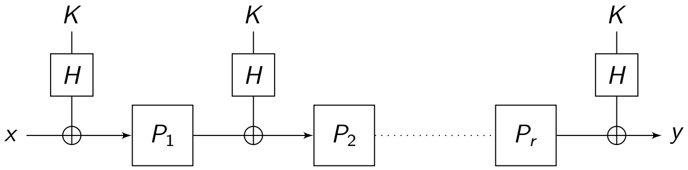
$\rightarrow$ indifferentiable from an ideal cipher with ℓ-bit keys for $r = 5$ ([ABD$^+$13] gives attacks up to 3 rounds)

Better bounds and less rounds than in this paper.

But the assumption about the key derivation is very strong and far from concrete designs (the key-schedule is often invertible)

The [ABD+13] result

IEM with a key-derivation function modeled as a random oracle from $\{0,1\}^\ell$ to $\{0,1\}^n$ (that the adversary queries in a black-box way)



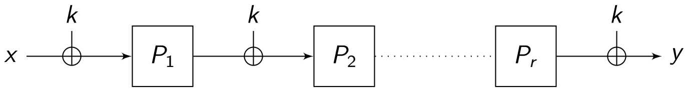
$\to$ indifferentiable from an ideal cipher with ℓ-bit keys for $r = 5$ ([ABD+13] gives attacks up to 3 rounds)

Better bounds and less rounds than in this paper.

But the assumption about the key derivation is very strong and far from concrete designs (the key-schedule is often invertible)

Our approach

We consider the IEM cipher with a single key:



The trivial attack on independent keys does not apply → is it indiff. from an ideal cipher for sufficiently many rounds ?

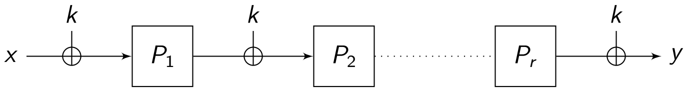
Main Result

The single-key IEM with $r = 12$ rounds is indifferentiable from an ideal cipher with n-bit blocks and n-bit keys

Also holds when using invertible permutations γ_i for the key derivation (no cryptographic assumption needed).

Our approach

We consider the IEM cipher with a single key:



The trivial attack on independent keys does not apply $\rightarrow$ is it indiff. from an ideal cipher for sufficiently many rounds ?

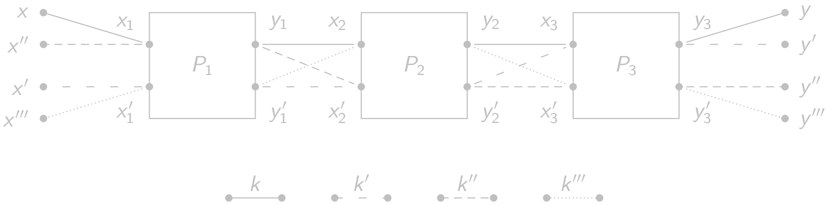
Main Result

The single-key IEM with $r = 12$ rounds is indifferentiable from an ideal cipher with n-bit blocks and n-bit keys

Also holds when using invertible permutations γ_i for the key derivation (no cryptographic assumption needed).

Outline

An attack for 3 rounds

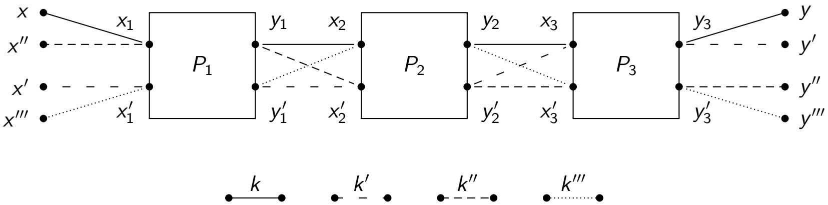


One can (easily) find (x, x', x'', x'''), (y, y', y'', y''') and (k, k', k'', k''') such that $y = \mathtt{IEM}^{(P_1,P_2,P_3)}(k, x)$, etc. and:

$$\begin{cases} k \oplus k' \oplus k'' \oplus k''' = 0 \\ x \oplus x' \oplus x'' \oplus x''' = 0 \\ y \oplus y' \oplus y'' \oplus y''' = 0 \end{cases} .$$

Finding such values can be showed to be hard for an ideal cipher.

An attack for 3 rounds

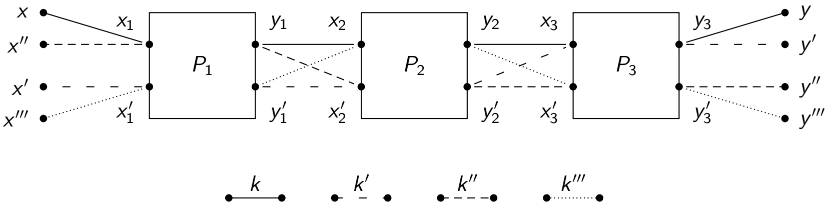


One can (easily) find (x, x', x'', x'''), (y, y', y'', y''') and (k, k', k'', k''') such that $y = \texttt{IEM}^{(P_1,P_2,P_3)}(k, x)$, etc. and:

$$\begin{cases} k \oplus k' \oplus k'' \oplus k''' = 0 \\ x \oplus x' \oplus x'' \oplus x''' = 0 \\ y \oplus y' \oplus y'' \oplus y''' = 0 \end{cases}.$$

Finding such values can be showed to be hard for an ideal cipher.

An attack for 3 rounds



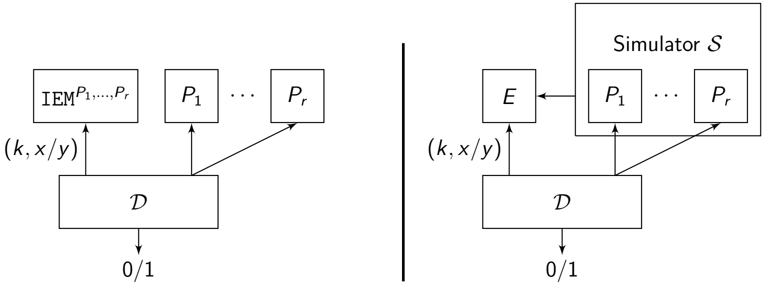
One can (easily) find (x, x', x'', x'''), (y, y', y'', y''') and (k, k', k'', k''') such that $y = \mathtt{IEM}^{(P_1,P_2,P_3)}(k, x)$, etc. and:

$$\begin{cases} k \oplus k' \oplus k'' \oplus k''' = 0 \\ x \oplus x' \oplus x'' \oplus x''' = 0 \\ y \oplus y' \oplus y'' \oplus y''' = 0 \end{cases} .$$

Finding such values can be showed to be hard for an ideal cipher.

Outline

Reminder: the indifferentiability setting



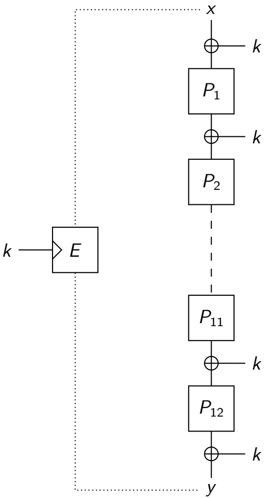
Simulation: general strategy

The simulator must return answers that are coherent with what the distinguisher can obtain from the ideal cipher E, i.e.:

$$\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$$

For this, the simulator must adapt at least one permutation to "match" what is given by the ideal cipher.

The general strategy is close to the one used for the indifferentiability of the Feistel permutation [CPS08, HKT11].



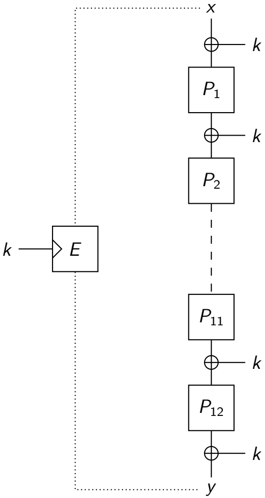
Simulation: general strategy

The simulator must return answers that are coherent with what the distinguisher can obtain from the ideal cipher E, i.e.:

$$\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$$

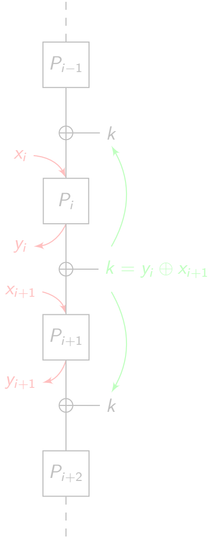
For this, the simulator must adapt at least one permutation to "match" what is given by the ideal cipher.

The general strategy is close to the one used for the indifferentiability of the Feistel permutation [CPS08, HKT11].



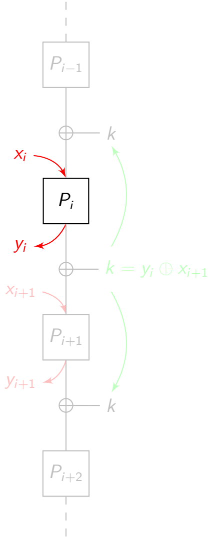
Simulation: general strategy

- the simulator maintains an history for each simulated permutation P_i
- the simulator detects and completes "partial chains" = queries to two adjacent perm. $P_i(x_i) = y_i$ and $P_{i+1}(x_{i+1}) = y_{i+1}$
- for any partial chain the key is uniquely defined: $k = y_i \oplus x_{i+1}$
- queries to any two consecutive permutations uniquely define the computations path in the construction (not true for independent keys!)



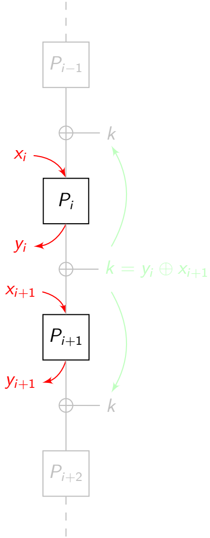
Simulation: general strategy

- the simulator maintains an history for each simulated permutation P_i
- the simulator detects and completes "partial chains" = queries to two adjacent perm. $P_i(x_i) = y_i$ and $P_{i+1}(x_{i+1}) = y_{i+1}$
- for any partial chain the key is uniquely defined: $k = y_i \oplus x_{i+1}$
- queries to any two consecutive permutations uniquely define the computations path in the construction (not true for independent keys!)



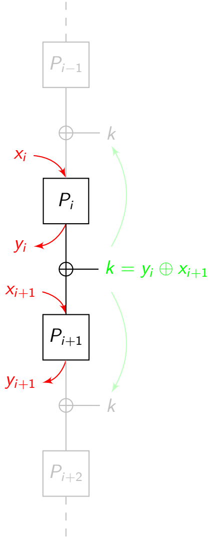
Simulation: general strategy

- the simulator maintains an history for each simulated permutation P_i
- the simulator detects and completes "partial chains" = queries to two adjacent perm. $P_i(x_i) = y_i$ and $P_{i+1}(x_{i+1}) = y_{i+1}$
- for any partial chain the key is uniquely defined: $k = y_i \oplus x_{i+1}$
- queries to any two consecutive permutations uniquely define the computations path in the construction (not true for independent keys!)



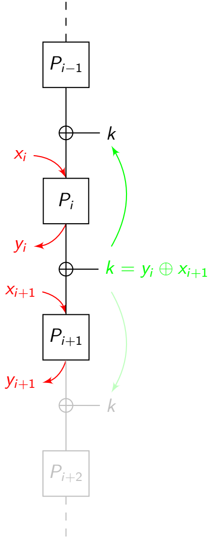
Simulation: general strategy

- the simulator maintains an history for each simulated permutation P_i
- the simulator detects and completes "partial chains" = queries to two adjacent perm. $P_i(x_i) = y_i$ and $P_{i+1}(x_{i+1}) = y_{i+1}$
- for any partial chain the key is uniquely defined: $k = y_i \oplus x_{i+1}$
- queries to any two consecutive permutations uniquely define the computations path in the construction (not true for independent keys!)



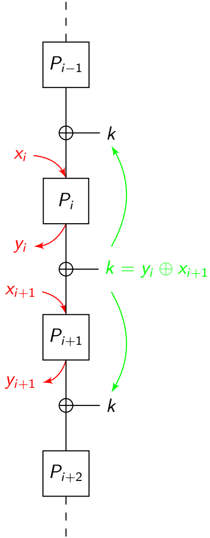
Simulation: general strategy

- the simulator maintains an history for each simulated permutation P_i
- the simulator detects and completes "partial chains" = queries to two adjacent perm. $P_i(x_i) = y_i$ and $P_{i+1}(x_{i+1}) = y_{i+1}$
- for any partial chain the key is uniquely defined: $k = y_i \oplus x_{i+1}$
- queries to any two consecutive permutations uniquely define the computations path in the construction (not true for independent keys!)

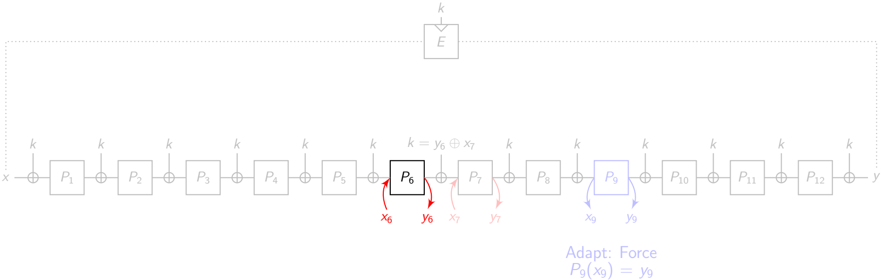


Simulation: general strategy

- the simulator maintains an history for each simulated permutation P_i
- the simulator detects and completes "partial chains" = queries to two adjacent perm. $P_i(x_i) = y_i$ and $P_{i+1}(x_{i+1}) = y_{i+1}$
- for any partial chain the key is uniquely defined: $k = y_i \oplus x_{i+1}$
- queries to any two consecutive permutations uniquely define the computations path in the construction (not true for independent keys!)

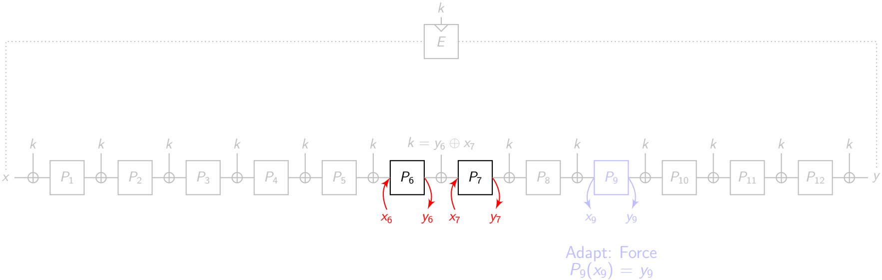


Completing a partial chain



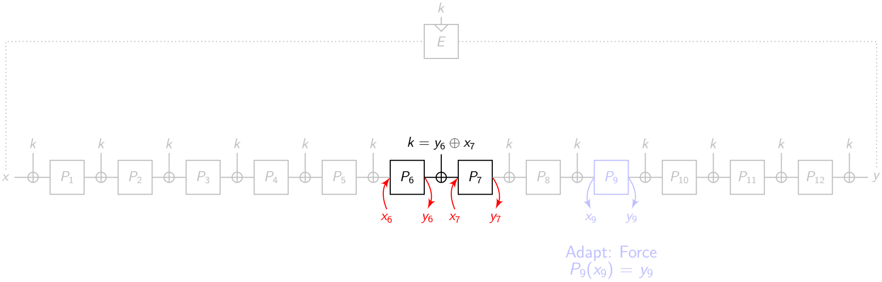
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



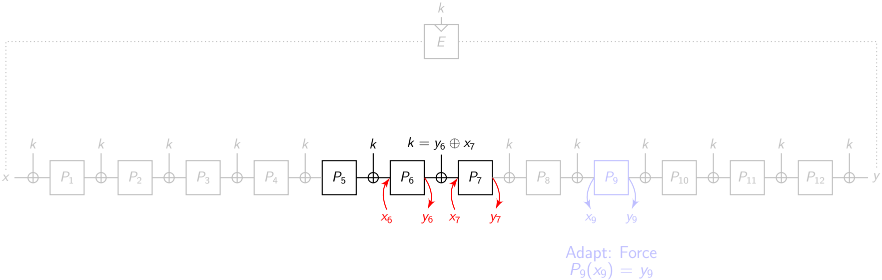
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



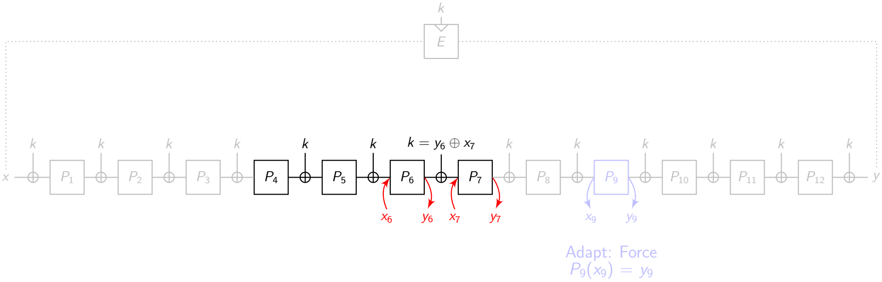
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



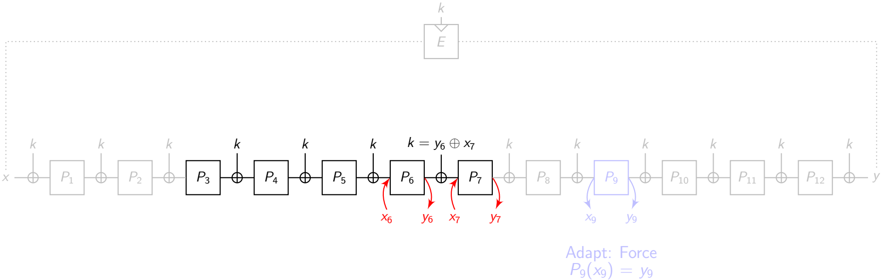
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\dots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



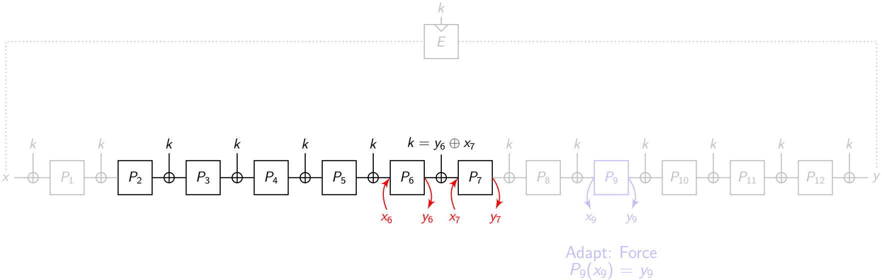
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



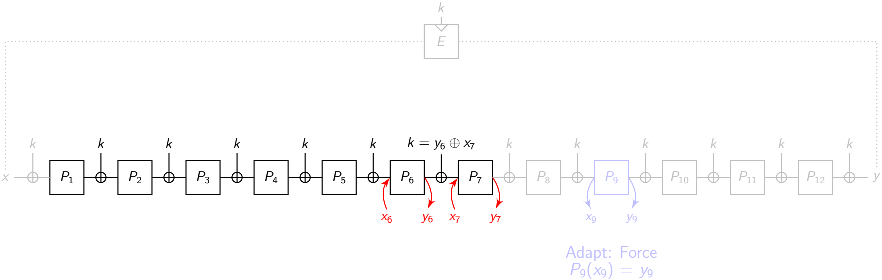
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



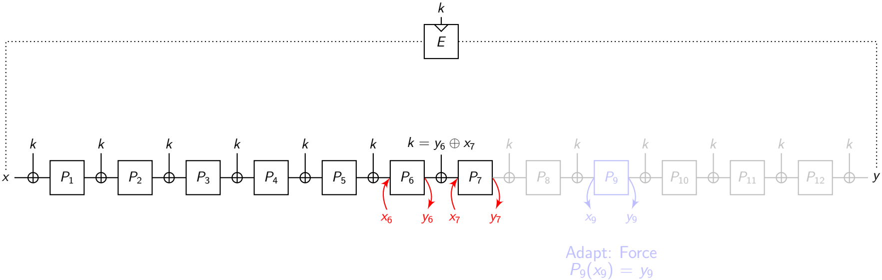
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k, x) = E(k, x)$.

Completing a partial chain



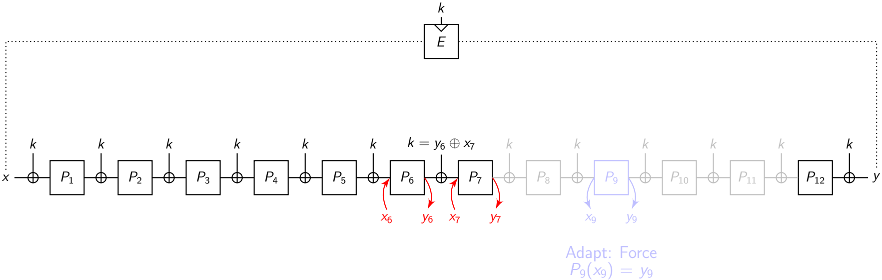
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathrm{IEM}^{P_1,\dots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



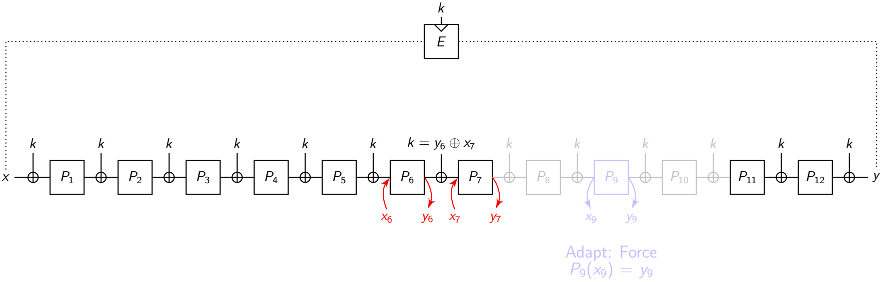
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



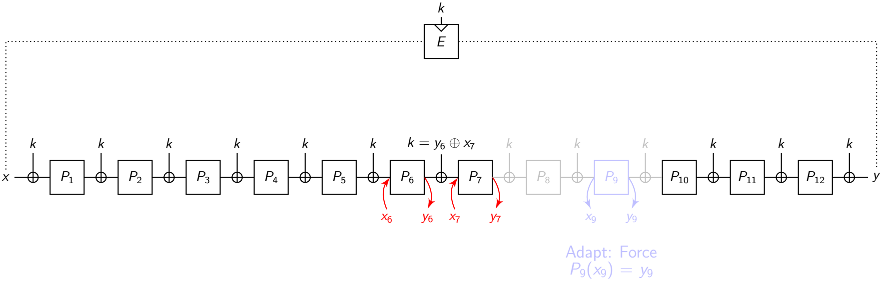
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



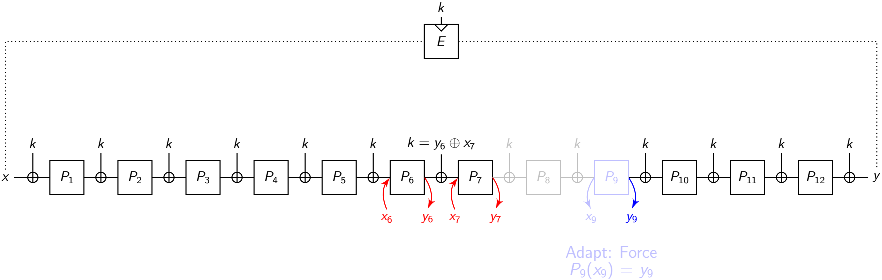
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k, x) = E(k, x)$.

Completing a partial chain



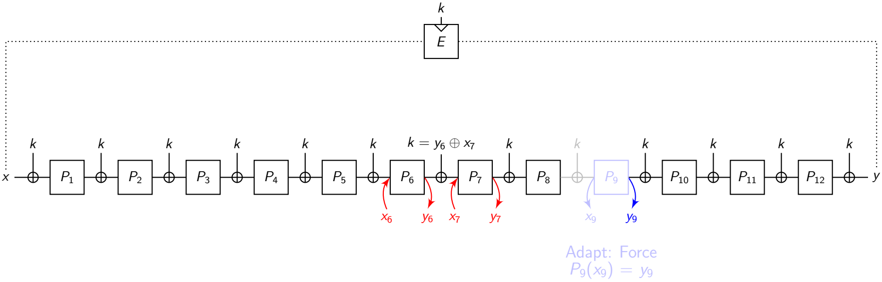
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



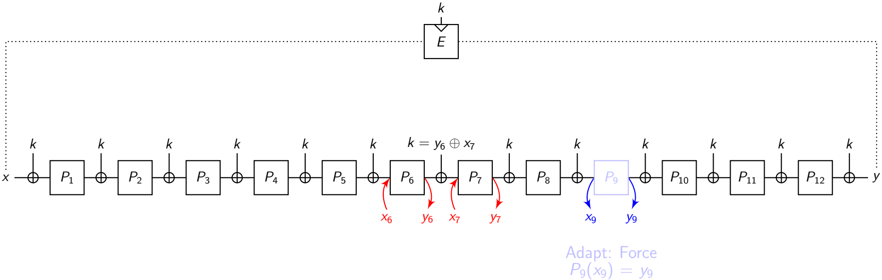
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



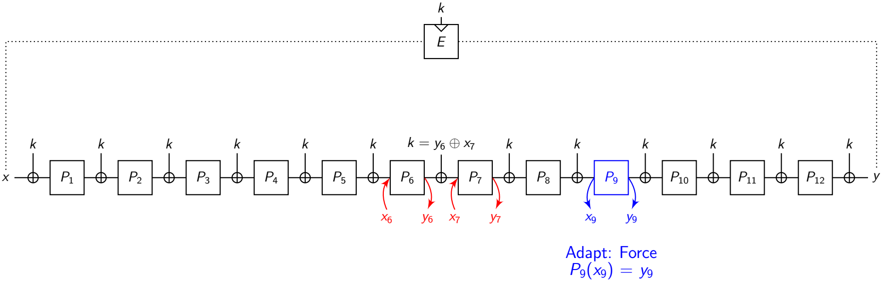
- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

Completing a partial chain



- when detecting a partial chain, $\mathcal{S}$ first completes the chain backward and forward randomly
- it makes a call to E to "wrap around"
- it forces $P_9(x_9) = y_9$ which ensures that $\mathtt{IEM}^{P_1,\ldots,P_{12}}(k,x) = E(k,x)$.

What could go wrong during simulation

Two problems to deal with:

1. complexity of the simulator:
   - completing a partial chain creates new chains, which must be completed, creating new partial chains, etc.
   - ⇒ potential blow-up of the number of chains completed by the simulator
   - but the simulator must be polynomial-time!
2. impossibility to adapt:
   - when the simulator wants to adapt a chain by forcing $P_i(x_i) = y_i$, it might happen that P_i was already defined for x_i or y_i
   - ⇒ the simulator cannot remain coherent with E!

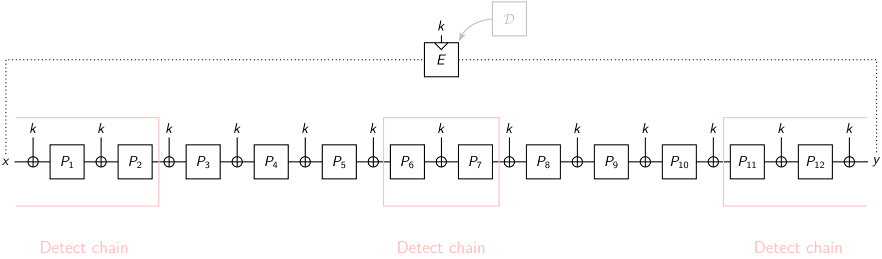
What could go wrong during simulation

Two problems to deal with:

1. complexity of the simulator:
   - completing a partial chain creates new chains, which must be completed, creating new partial chains, etc.
   - $\Rightarrow$ potential blow-up of the number of chains completed by the simulator
   - but the simulator must be polynomial-time!
2. impossibility to adapt:
   - when the simulator wants to adapt a chain by forcing $P_i(x_i) = y_i$, it might happen that P_i was already defined for x_i or y_i
   - $\Rightarrow$ the simulator cannot remain coherent with E!

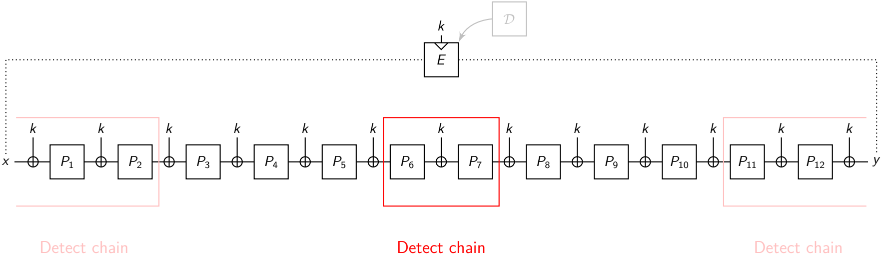
Bounding the simulator's complexity

- the simulator only detects and completes partial chains at very specific places:
  - central chains: queries to (P_6, P_7)
  - external chains: queries to $(P_1, P_2, P_{11}, P_{12})$ that matches E
- an external chain can be created only if the distinguisher has made the corresponding query to E
  $\rightarrow$ only q of them will be completed, which avoids a recursive blow-up of the simulator



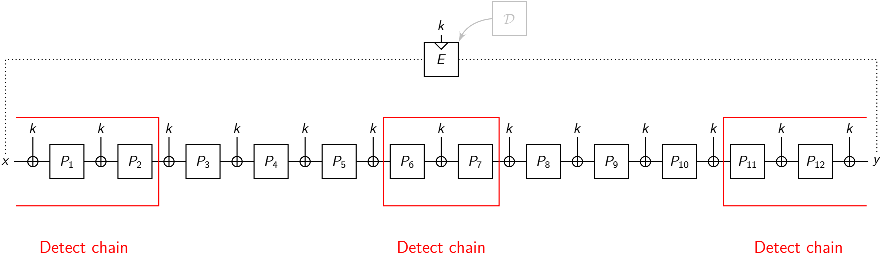
Bounding the simulator's complexity

- the simulator only detects and completes partial chains at very specific places:
  - central chains: queries to (P_6, P_7)
  - external chains: queries to $(P_1, P_2, P_{11}, P_{12})$ that matches E
- an external chain can be created only if the distinguisher has made the corresponding query to E
  $\rightarrow$ only q of them will be completed, which avoids a recursive blow-up of the simulator



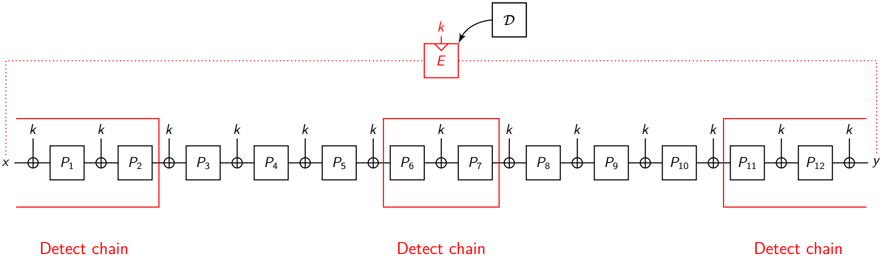
Bounding the simulator's complexity

- the simulator only detects and completes partial chains at very specific places:
  - central chains: queries to (P_6, P_7)
  - external chains: queries to $(P_1, P_2, P_{11}, P_{12})$ that matches E
- an external chain can be created only if the distinguisher has made the corresponding query to E
  $\rightarrow$ only q of them will be completed, which avoids a recursive blow-up of the simulator



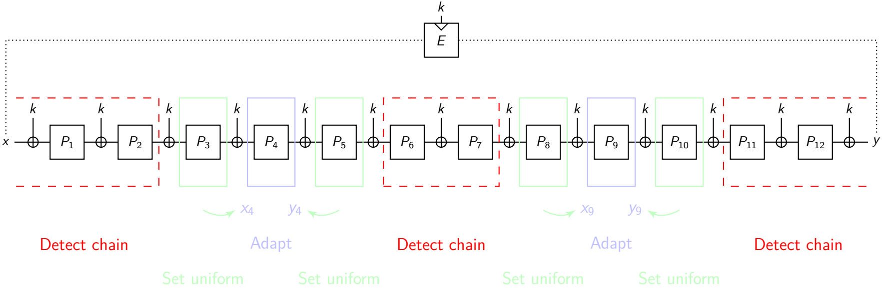
Bounding the simulator's complexity

- the simulator only detects and completes partial chains at very specific places:
    - central chains: queries to (P_6, P_7)
    - external chains: queries to $(P_1, P_2, P_{11}, P_{12})$ that matches E
- an external chain can be created only if the distinguisher has made the corresponding query to E
  $\rightarrow$ only q of them will be completed, which avoids a recursive blow-up of the simulator



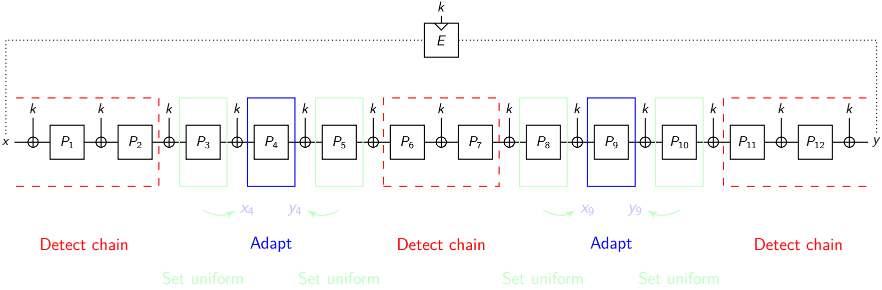
Ensuring that the simulator can always adapt

- chains are always adapted at P_4 or P_9
- adaptation rounds are surrounded by buffer rounds whose answers are drawn at random just before adapting
- the values (x_4, y_4) or (x_9, y_9) used to adapt P_4 or P_9 are random
  $\Rightarrow$ in the history of the simulator only with negl. probability



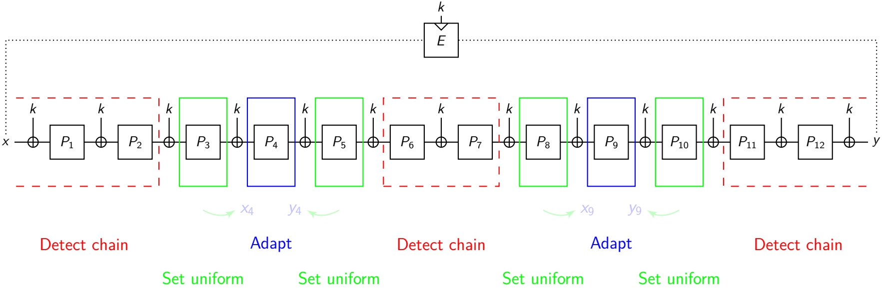
Ensuring that the simulator can always adapt

- chains are always adapted at P_4 or P_9
- adaptation rounds are surrounded by buffer rounds whose answers are drawn at random just before adapting
- the values (x_4, y_4) or (x_9, y_9) used to adapt P_4 or P_9 are random
  $\Rightarrow$ in the history of the simulator only with negl. probability



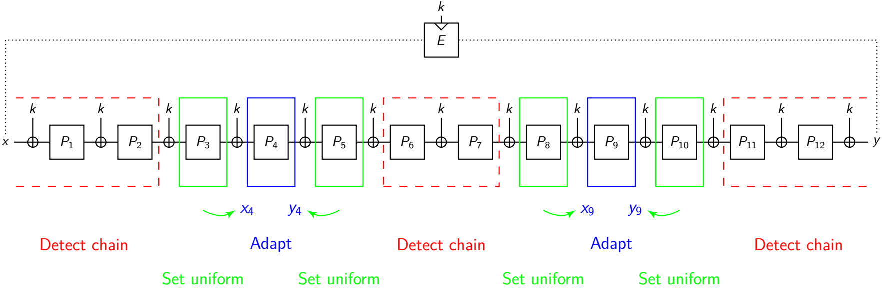
Ensuring that the simulator can always adapt

- chains are always adapted at P_4 or P_9
- adaptation rounds are surrounded by buffer rounds whose answers are drawn at random just before adapting
- the values (x_4, y_4) or (x_9, y_9) used to adapt P_4 or P_9 are random $\Rightarrow$ in the history of the simulator only with negl. probability



Ensuring that the simulator can always adapt

- chains are always adapted at P_4 or P_9
- adaptation rounds are surrounded by buffer rounds whose answers are drawn at random just before adapting
- the values (x_4, y_4) or (x_9, y_9) used to adapt P_4 or P_9 are random $\Rightarrow$ in the history of the simulator only with negl. probability



Conclusion

Main result

The single-key IEM cipher with 12 rounds is indifferentiable from an ideal cipher with n-bit keys.

Interpretation of the result:

- shows that the general strategy of building block ciphers from SPNs is sound and may even yield something close to an ideal cipher
- says little about concrete block ciphers: e.g. the permutations P_1, $\ldots$, P_{10} of AES-128 are too simple and not independent
- gives heuristic insurance for e.g. an IEM cipher where the P_i's are instantiated with AES used with fixed keys

Conclusion

Main result

The single-key IEM cipher with 12 rounds is indifferentiable from an ideal cipher with n-bit keys.

Interpretation of the result:

- shows that the general strategy of building block ciphers from SPNs is sound and may even yield something close to an ideal cipher
- says little about concrete block ciphers: e.g. the permutations P_1, ..., P_{10} of AES-128 are too simple and not independent
- gives heuristic insurance for e.g. an IEM cipher where the P_i's are instantiated with AES used with fixed keys

Open problems

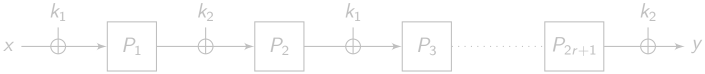
1. exact number of rounds for indifferentiability?
   - The indifferentiability proof requires 12 rounds...
     but the best attack is only on 3 rounds.

Conjecture

The single-key IEM with $3 < r < 12$ rounds is indifferentiable from an ideal cipher with n-bit keys

   - $r = 4$ may well be sufficient
     (we explain which obstacles appear already for $r = 8$ in the full paper)

2. construction with $2n$-bit keys? (or more generally tn-bit keys with $t > 1$)



Open problems

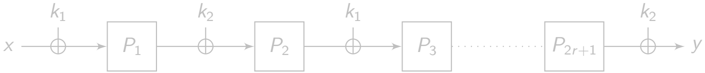
1. exact number of rounds for indifferentiability?
   - The indifferentiability proof requires 12 rounds...
     but the best attack is only on 3 rounds.

Conjecture

The single-key IEM with $3 < r < 12$ rounds is indifferentiable from an ideal cipher with n-bit keys

   - $r = 4$ may well be sufficient
     (we explain which obstacles appear already for $r = 8$ in the full paper)

2. construction with $2n$-bit keys? (or more generally tn-bit keys with $t > 1$)



Open problems

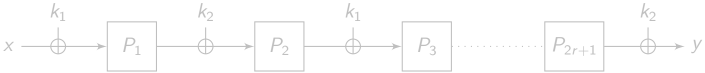
1. exact number of rounds for indifferentiability?
   - The indifferentiability proof requires 12 rounds...
     but the best attack is only on 3 rounds.

Conjecture

The single-key IEM with $3 < r < 12$ rounds is indifferentiable from an ideal cipher with n-bit keys

   - $r = 4$ may well be sufficient
     (we explain which obstacles appear already for $r = 8$ in the full paper)

2. construction with $2n$-bit keys? (or more generally tn-bit keys with $t > 1$)



Open problems

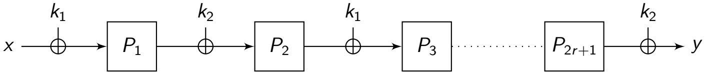
1. exact number of rounds for indifferentiability?
   - The indifferentiability proof requires 12 rounds...
     but the best attack is only on 3 rounds.

Conjecture

The single-key IEM with $3 < r < 12$ rounds is indifferentiable from an ideal cipher with n-bit keys

   - $r = 4$ may well be sufficient
     (we explain which obstacles appear already for $r = 8$ in the full paper)

2. construction with $2n$-bit keys? (or more generally tn-bit keys with $t > 1$)



The end. . .

Thanks for your attention!
Comments or questions?

References I

Elena Andreeva, Andrey Bogdanov, Yevgeniy Dodis, Bart Mennink, and John P. Steinberger.

On the Indifferentiability of Key-Alternating Ciphers.

In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 (Proceedings, Part I)*, volume 8042 of *Lecture Notes in Computer Science*, pages 531–550. Springer, 2013.

Full version available at `http://eprint.iacr.org/2013/061`.

Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, François-Xavier Standaert, John P. Steinberger, and Elmar Tischhauser.

Key-Alternating Ciphers in a Provable Setting: Encryption Using a Small Number of Public Permutations - (Extended Abstract).

In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 45–62. Springer, 2012.

References II

Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin.
The Random Oracle Model and the Ideal Cipher Model Are Equivalent.
In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2008.

Shan Chen and John Steinberger.
Tight Security Bounds for Key-Alternating Ciphers.
IACR Cryptology ePrint Archive, Report 2013/222, 2013.
Available at `http://eprint.iacr.org/2013/222`.

Shimon Even and Yishay Mansour.
A Construction of a Cipher from a Single Pseudorandom Permutation.
Journal of Cryptology, 10(3):151–162, 1997.

References III

Thomas Holenstein, Robin Künzler, and Stefano Tessaro.

The Equivalence of the Random Oracle Model and the Ideal Cipher Model, Revisited.

In Lance Fortnow and Salil P. Vadhan, editors, *Symposium on Theory of Computing - STOC 2011*, pages 89–98. ACM, 2011.

Full version available at `http://arxiv.org/abs/1011.1264`.

Rodolphe Lampe, Jacques Patarin, and Yannick Seurin.

An Asymptotically Tight Security Analysis of the Iterated Even-Mansour Cipher.

In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 278–295. Springer, 2012.

References IV

Ueli M. Maurer, Renato Renner, and Clemens Holenstein.

Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology.

In Moni Naor, editor, *Theory of Cryptography Conference- TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

John Steinberger.

Counting solutions to additive equations in random sets.

arXiv Report 1309.5582, 2013.

Available at http://arxiv.org/abs/1309.5582.