# Key-Alternating Ciphers in a Provable Setting: Encryption Using a Small Number of Public Permutations

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

EUROCRYPT 2012

# Outline

# Outline

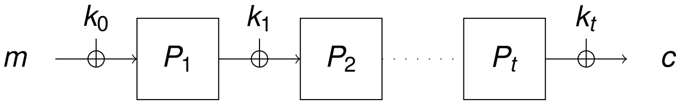


DTU

# The Cipher

## The Topic

A key-alternating cipher




## Example

Most prominent example: AES

Many others exist.

# A natural question

**Question**

Is this a good way of building block ciphers?

# A natural question

**Question**

Is this a good way of building block ciphers?

More precisely:

**Question**

Is there a generic way to break all of them?

# A natural question

**Question**

Is this a good way of building block ciphers?

More precisely:

**Question**

Is there a generic way to break all of them?

**Answer**

We do not know!

Only one round studied: Even-Mansour '91

# A natural question

**Answer**

We do not know!

Very surprising!

- cf. progress in provable security
- cf. generic group model
- cf. SHA-3 competition

# SHA-3

**Folklore**

We are much more confident with designing block ciphers than with designing hash functions.

**Example**

DES is still okay, MD4/MD5 and SHA-1: not really

# SHA-3

**Folklore**

We are much more confident with designing block ciphers than with designing hash functions.

**Example**

DES is still okay, MD4/MD5 and SHA-1: not really

When it comes to provable security: This is different.

**Provable Security**

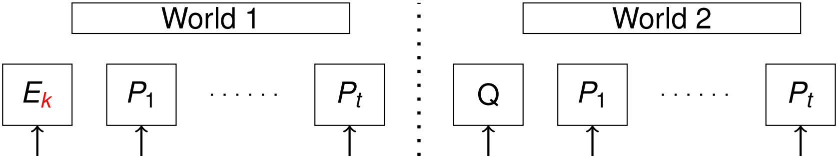
All SHA-3 finalist come with a proof in the idealized model. No AES finalist had one.

# Outline

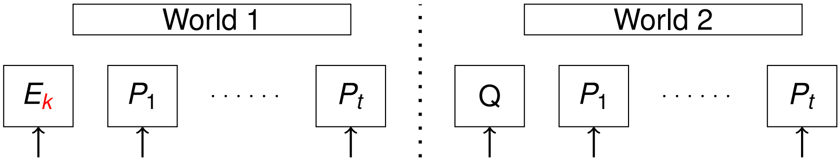
# The Setting

## The Setting

- Ideal round functions
- Information theoretical adversary
- Two worlds



# The Result




## Theorem (informal)

*No adversary can distinguish the two worlds with less than*

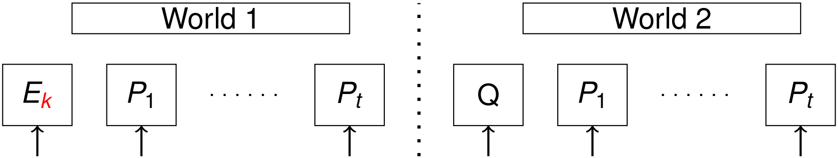
- $2^{n/2}$ *queries for one round (Even-Mansour)*
- $2^{2n/3}$ *queries for more than one round*
- $2^{3n/4}$ *queries for more than two rounds*

# Outline

# A Proof Outline

## Initial Game

- Sample permutations $P_i$ and $Q$ uniform at random
- Choose random keys
- Goal of the adversary $A$: distinguish the worlds



# Lazy Sampling

Start with empty lists for $P_i$ and $E$.

**Upon a query to $P_i$ (or $E$):**

- Select all $P_i$ uniform at random (among all permutations consistent with previous queries).
- Construct $E$.
- Answer the query accordingly.
- Update lists.

Goal of the adversary $A$: distinguish the worlds

Clear: Results in the same distribution.

# The Hybrid

We change the game *a bit*. Lazy sampling: Start with empty lists for $P_i$ and $E$.

# The Hybrid

We change the game *a bit*. Lazy sampling: Start with empty lists for $P_i$ and $E$.

**Modified Game: Upon a query to $P_i$ (or $E$):**

- Select a random answer $y$ (maintaining $P_i$ (or $E$) as a permutation)
- Check consistency
- If consistent: Output $y$ and update lists.
- If inconsistent: crash!

Goal of the adversary $A$: distinguish the worlds

# The Hybrid

We change the game *a bit*. Lazy sampling: Start with empty lists for $P_i$ and $E$.

**Modified Game: Upon a query to $P_i$ (or $E$):**

- Select a random answer $y$ (maintaining $P_i$ (or $E$) as a permutation)
- Check consistency
- If consistent: Output $y$ and update lists.
- If inconsistent: crash!

Goal of the adversary $A$: distinguish the worlds

**Consistency**

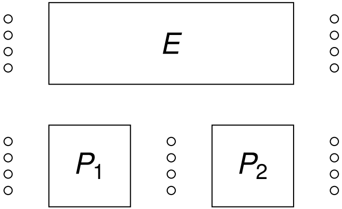
$P_i$ and $E_k$ are consistent iff

$$E_k(x) = P_t(\dots P_2(P_1(x \oplus k_0) \oplus k_1)\dots) \oplus k_t$$

# The Hybrid: In a Picture

**For Simplicity**

Only $n = 2$ and zero keys.

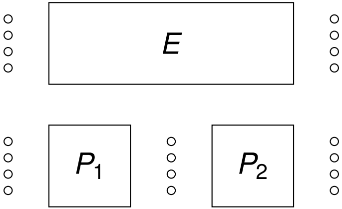



List of queries:

# The Hybrid: In a Picture

**For Simplicity**

Only $n = 2$ and zero keys.



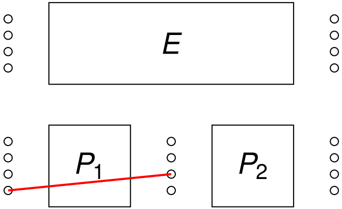

List of queries:

$$P_1(0) =$$

# The Hybrid: In a Picture

## For Simplicity

Only $n = 2$ and zero keys.



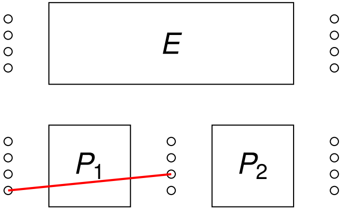

List of queries:

$$P_1(0) = 1$$

# The Hybrid: In a Picture

**For Simplicity**

Only $n = 2$ and zero keys.



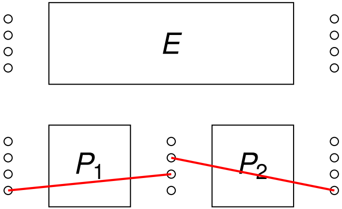

List of queries:

$$P_1(0) = 1 \quad P_2(2) =$$

# The Hybrid: In a Picture

## For Simplicity

Only $n = 2$ and zero keys.



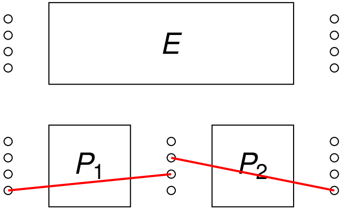

List of queries:

$$P_1(0) = 1 \quad P_2(2) = 0$$

# The Hybrid: In a Picture

## For Simplicity

Only $n = 2$ and zero keys.



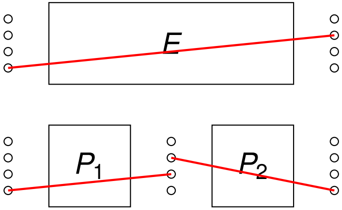

List of queries:

$$P_1(0) = 1 \quad P_2(2) = 0 \quad E(0) =$$

# The Hybrid: In a Picture

## For Simplicity

Only $n = 2$ and zero keys.



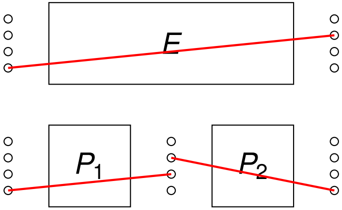

List of queries:

$$P_1(0) = 1 \quad P_2(2) = 0 \quad E(0) = 2$$

# The Hybrid: In a Picture

## For Simplicity

Only $n = 2$ and zero keys.



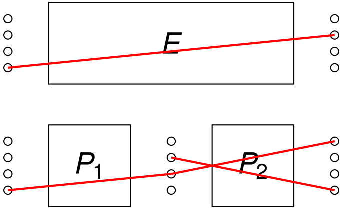

List of queries:

$$P_1(0) = 1 \quad P_2(2) = 0 \quad E(0) = 2 \quad P_2(1) =$$

# The Hybrid: In a Picture

**For Simplicity**

Only $n = 2$ and zero keys.



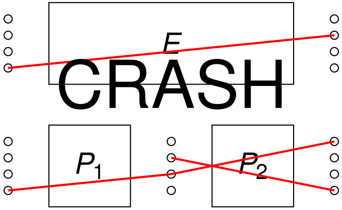

List of queries:

$$P_1(0) = 1 \quad P_2(2) = 0 \quad E(0) = 2 \quad P_2(1) = 3$$

# The Hybrid: In a Picture

**For Simplicity**

Only $n = 2$ and zero keys.




List of queries:

$$P_1(0) = 1 \quad P_2(2) = 0 \quad E(0) = 2 \quad P_2(1) = 3$$

# Almost Done?

## Two Steps To Go

1. Show that one cannot win in the modified game.
2. Show that the modified game is only *slightly* different.

# Almost Done?

## Two Steps To Go

1. Show that one cannot win in the modified game.
2. Show that the modified game is only *slightly* different.

## Step 1: Easy

Sketch on the next slides.

# Almost Done?

## Two Steps To Go

1. Show that one cannot win in the modified game.
2. Show that the modified game is only *slightly* different.

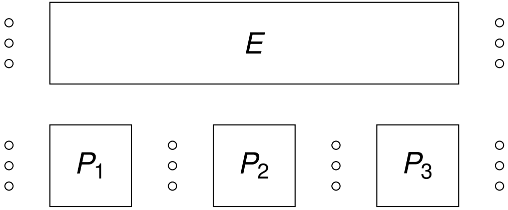
## Step 1: Easy

Sketch on the next slides.

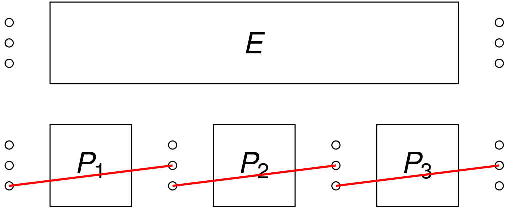
## Step 2: Not so easy

Quite involved and technical: See paper.

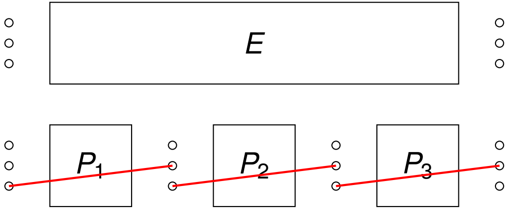
# On Step 2: The Modified Game is different



# On Step 2: The Modified Game is different

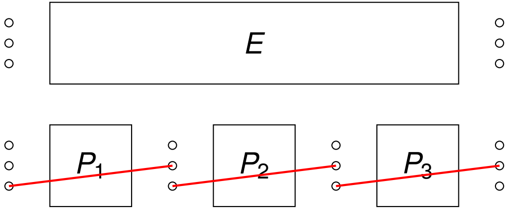


# On Step 2: The Modified Game is different




$$E(0) = ?$$

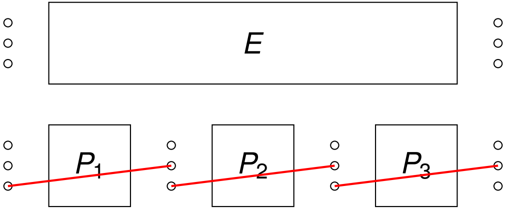
# On Step 2: The Modified Game is different




$$E(0) = ?$$

- Modified Game:
  $\Pr(E(0) = 0) = \Pr(E(0) = 1) = \Pr(E(0) = 2) = 1/3$

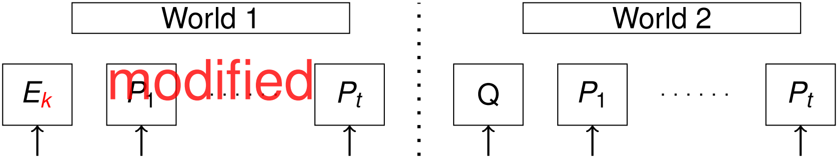
# On Step 2: The Modified Game is different




$$E(0) = ?$$

- Modified Game:
  $\Pr(E(0) = 0) = \Pr(E(0) = 1) = \Pr(E(0) = 2) = 1/3$
- Original Game: 8 possibilities

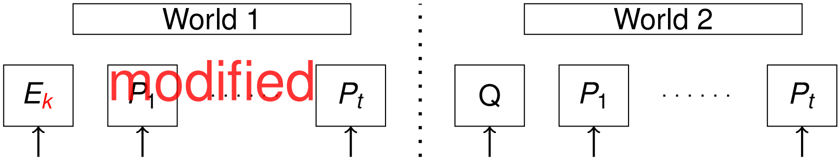
# Step 1




**First observation**

As long as the oracle does not crash, both worlds are the same.

# Step 1




**First observation**

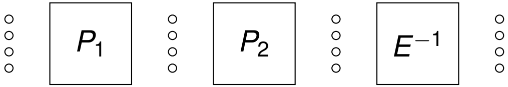
As long as the oracle does not crash, both worlds are the same.

What is the probability for a crash?

# Step 1, continued

**Question**

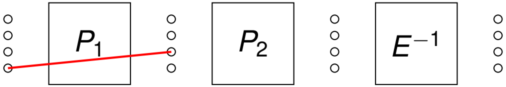
What is the probability for a crash?



# Step 1, continued

**Question**

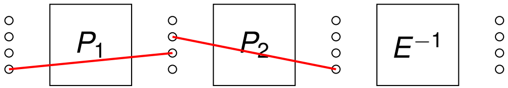
What is the probability for a crash?



# Step 1, continued

**Question**

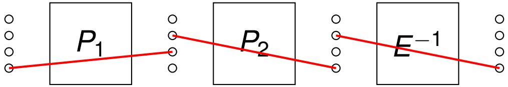
What is the probability for a crash?



# Step 1, continued

**Question**

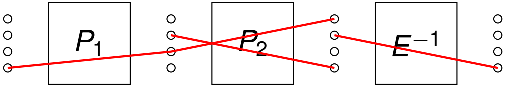
What is the probability for a crash?



# Step 1, continued

**Question**

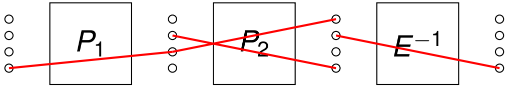
What is the probability for a crash?



# Step 1, continued

## Question

What is the probability for a crash?




## A Crash

A sequence of queries, connected in all but one positions.

# Step 1, continued

**A Crash**

A sequence of queries, connected in all but one positions.

# Step 1, continued

**A Crash**

A sequence of queries, connected in all but one positions.

Number of sequences:

$$\leq q^{t+1}$$

# Step 1, continued

**A Crash**

A sequence of queries, connected in all but one positions.

Number of sequences:

$$\leq q^{t+1}$$

Prob for a sequences to be connected in all but one positions:

$$\leq (t+1)2^{-tn}$$

# Step 1, continued

**A Crash**

A sequence of queries, connected in all but one positions.

Number of sequences:

$$\leq q^{t+1}$$

Prob for a sequences to be connected in all but one positions:

$$\leq (t+1)2^{-tn}$$

Thus

$$\Pr(\text{crash}) \leq \frac{(t+1)q^{t+1}}{2^{tn}}$$

# Step 1, continued

## A Crash

A sequence of queries, connected in all but one positions.

Number of sequences:

$$\leq q^{t+1}$$

Prob for a sequences to be connected in all but one positions:

$$\leq (t+1)2^{-tn}$$

Thus

$$\Pr(\text{crash}) \leq \frac{(t+1)q^{t+1}}{2^{tn}}$$

## Informal

$$q \approx 2^{\frac{t}{t+1}n}$$

# The precise statement

Notation:

- $n$ block size
- $q$ number of queries

**Theorem**

*Let $N = 2^n$ and let $q = N^{\frac{t}{t+1}}/Z$ for some $Z \geq 1$. Then, for any $t \geq 1$, and assuming $q < N/100$, we have*

$$\mathbf{Adv}^{\mathrm{PRP}}_{E,N,t}(q) \leq \frac{4.3q^3t}{N^2} + \frac{t+1}{Z^t}.$$

For $t \geq 2$ this implies $q \approx 2^{2n/3}$.

# Outline

# Interpreting The Result

**Theorem (informal)**

*With idealized permutations as round function, the key-alternating cipher is secure.*

What does this mean? For a concrete cipher?

# Interpreting The Result

**Theorem (informal)**

*With idealized permutations as round function, the key-alternating cipher is secure.*

What does this mean? For a concrete cipher?

**Generic Attacks**

If you want to break the cipher, you have to use special properties of the permutations.

# Interpreting The Result

**Theorem (informal)**

*With idealized permutations as round function, the key-alternating cipher is secure.*

What does this mean? For a concrete cipher?

**Generic Attacks**

If you want to break the cipher, you have to use special properties of the permutations.

**For a Concrete Instance**

It does not mean anything.

Any practical instance is, well practical. Thus not ideal.

# Outline

# Further Results

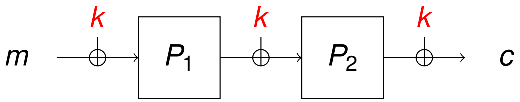
More in the paper:

- Study the expected resistance against linear cryptanalysis
- A concrete proposal using AES

# A Concrete Proposal or How It Started

## A Proposal

A block cipher secure against related-key attacks?




($P_{1,2}$ is AES with fixed key.)

## Intuition

If $P_1$ and $P_2$ are complicated enough than related-key attacks do not work.

# Future Work

This work leaves many questions open:

- Improve the bound
- Get closer to actual constructions, e.g.
  - identical round keys
  - identical round functions
- New Constructions

# Future Work: Improve The Bound

**Conjecture**

The actual lower bound is

$$q \approx 2^{t/(t+1)n}$$

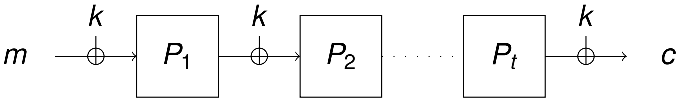
- This is actually the upper bound.
- Already improved to $2^{3/4n}$ for $t \geq 3$
- Challenging step: A bound that improves with $t$.

# The End

Thanks!

# Future Work: Closer to actual constructions I

Identical round keys:



# Future Work: Closer to actual constructions II

Identical round functions: