Low-Memory Attacks against 2-Round Even-Mansour using the 3-XOR Problem

Gaëtan Leurent, Ferdinand Sibleyras

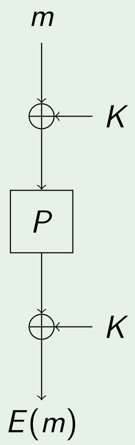
Inria, France

Crypto 2019

1-Round Even-Mansour

Most-Simple permutation-based block cipher.
Original by Even and Mansour, Asiacrypt 91.
Single-key by Dunkelman *et al.*, Eurocrypt 2012.

1EM

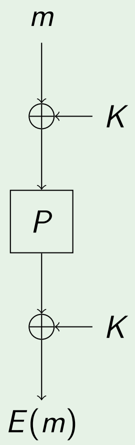


1-Round Even-Mansour

Most-Simple permutation-based block cipher.
Original by Even and Mansour, Asiacrypt 91.
Single-key by Dunkelman *et al.*, Eurocrypt 2012.

$$\left.\begin{array}{l} n\text{-bit to } n\text{-bit public permutation } P. \\ n\text{-bit secret key } K. \end{array}\right\} \text{secure block cipher } E.$$

1EM



1-Round Even-Mansour

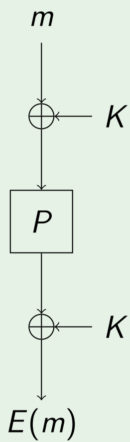
Most-Simple permutation-based block cipher.
Original by Even and Mansour, Asiacrypt 91.
Single-key by Dunkelman *et al.*, Eurocrypt 2012.

$$\left.\begin{array}{l} n\text{-bit to } n\text{-bit public permutation } P. \\ n\text{-bit secret key } K. \end{array}\right\} \text{secure block cipher } E.$$

D = number of calls to keyed E,
Q = number of calls to the public P,
1EM provable security up to $DQ \ll 2^n$.

$\Longrightarrow$ Security up to birthday bound $2^{n/2}$.

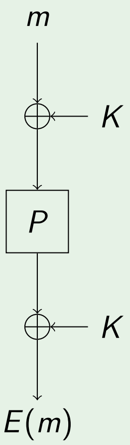
1EM



1-Round Even-Mansour

Cryptanalysis in $DQ = DT = 2^n$ originally by Daemen, Asiacrypt 91.

1EM



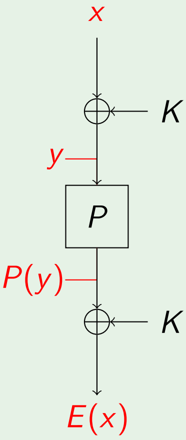
1-Round Even-Mansour

Cryptanalysis in $DQ = DT = 2^n$ originally by Daemen, Asiacrypt 91.

$\forall x, y \in \{0,1\}^n$,

$$x \oplus y = K \iff P(y) \oplus E(x) = K$$

1EM



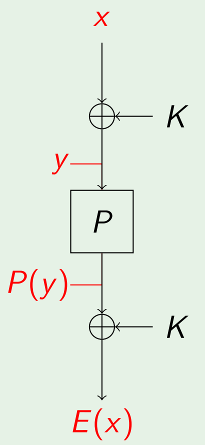
1-Round Even-Mansour

Cryptanalysis in $DQ = DT = 2^n$ originally by Daemen, Asiacrypt 91.

$\forall x, y \in \{0,1\}^n$,

$$\begin{aligned} x \oplus y = K &\iff P(y) \oplus E(x) = K \\ &\implies x \oplus E(x) \oplus y \oplus P(y) = 0 \end{aligned}$$

1EM



1-Round Even-Mansour

Cryptanalysis in $DQ = DT = 2^n$ originally by Daemen, Asiacrypt 91.

$\forall x, y \in \{0,1\}^n$,

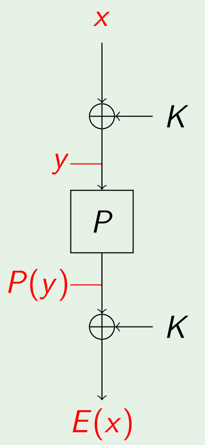
$$x \oplus y = K \iff P(y) \oplus E(x) = K$$
$$\implies x \oplus E(x) \oplus y \oplus P(y) = 0$$

Cryptanalysis via *n*-bit collision search

Let $f_0(x) = x \oplus E(x)$ and $f_1(y) = y \oplus P(y)$.
Find a collision between f_0 and f_1, guess $K = x \oplus y$.

$\implies$ No gap between the best proofs and attacks.

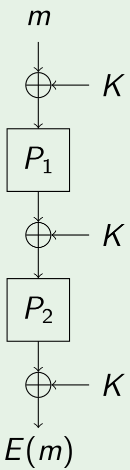
1EM



2-Round Even-Mansour

Extension by Bogdanov *et al.*, Eurocrypt 2012.
Keeps it simple and secure beyond birthday-bound.

2EM

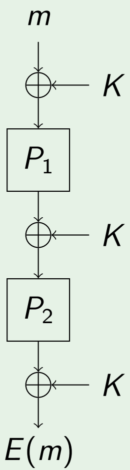


2-Round Even-Mansour

Extension by Bogdanov *et al.*, Eurocrypt 2012.
Keeps it simple and secure beyond birthday-bound.

Provably secure up to $2^{2n/3}$.
Best cryptanalysis time complexity: $T = 2^n/n$.

2EM



2-Round Even-Mansour

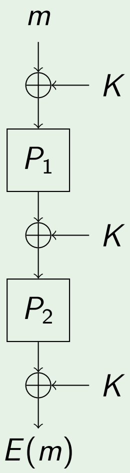
Extension by Bogdanov *et al.*, Eurocrypt 2012.
Keeps it simple and secure beyond birthday-bound.

Provably secure up to $2^{2n/3}$.
Best cryptanalysis time complexity: $T = 2^n/n$.

GAP

There remains a significant gap between the proof, $2^{2n/3}$, and the best attacks in $T = 2^n/n$.

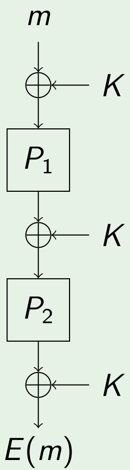
2EM



Our Approach

Best information theoretic attack trade-off: $DQ^2 = 2^{2n}$.
This matches the proof only in $D = Q = 2^{2n/3}$.
Best time complexity cryptanalysis in $T = 2^n/n$ but it uses also a lot of memory and/or online data!

2EM



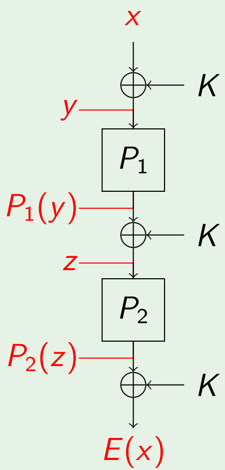
Our Approach

Best information theoretic attack trade-off: $DQ^2 = 2^{2n}$.
This matches the proof only in $D = Q = 2^{2n/3}$.
Best time complexity cryptanalysis in $T = 2^n/n$ but it uses also a lot of memory and/or online data!

In this work, we use the fact that:
$\forall x, y, z \in \{0,1\}^n$,

$$\begin{cases} x \oplus y & = K \\ P_1(y) \oplus z & = K \end{cases} \iff \begin{cases} x \oplus y & = K \\ P_1(y) \oplus z & = K \\ P_2(z) \oplus E(x) & = K \end{cases}$$

2EM



Our Approach

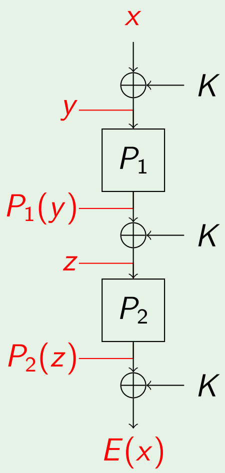
Best information theoretic attack trade-off: $DQ^2 = 2^{2n}$.
This matches the proof only in $D = Q = 2^{2n/3}$.
Best time complexity cryptanalysis in $T = 2^n/n$ but it uses also a lot of memory and/or online data!

In this work, we use the fact that:
$\forall x, y, z \in \{0,1\}^n$,

$$\begin{cases} x \oplus y & = K \\ P_1(y) \oplus z & = K \end{cases} \iff \begin{cases} x \oplus y & = K \\ P_1(y) \oplus z & = K \\ P_2(z) \oplus E(x) & = K \end{cases}$$

$$\implies \begin{cases} x & \oplus & y \oplus P_1(y) & \oplus & z & = 0 \\ x \oplus E(x) & \oplus & y & \oplus & P_2(z) & = 0 \end{cases}$$

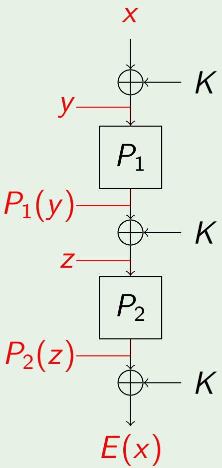
2EM



First result : A Link to the 3-XOR

$$\begin{cases} x \oplus y \oplus P_1(y) \oplus z = 0 \\ x \oplus E(x) \oplus y \oplus P_2(z) = 0 \end{cases}$$

2EM



First result : A Link to the 3-XOR

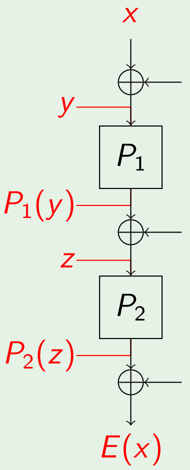
$$\begin{cases} x \oplus y \oplus P_1(y) \oplus z = 0 \\ x \oplus E(x) \oplus y \oplus P_2(z) = 0 \end{cases}$$

Cryptanalysis via the 3-XOR Problem with 2n-bit functions

$$\begin{aligned} f_0(x) &= x \;\|\; x \oplus E(x) \\ f_1(y) &= y \oplus P_1(y) \;\|\; y \\ f_2(z) &= z \;\|\; P_2(z) \end{aligned}$$

Solve the 3-XOR problem between f_0, f_1 and f_2.
Guess $K = x \oplus y$.

2EM



3-XOR Problem

Definition (Collision problem)

Given two functions f_0, f_1, find two inputs (x_0, x_1) such that $f_0(x_0) \oplus f_1(x_1) = 0$.

3-XOR Problem

Definition (Collision problem)

Given two functions f_0, f_1, find two inputs (x_0, x_1) such that $f_0(x_0) \oplus f_1(x_1) = 0$.

Definition (3-XOR problem)

Given three functions f_0, f_1, f_2, find three inputs (x_0, x_1, x_2) such that $f_0(x_0) \oplus f_1(x_1) \oplus f_2(x_2) = 0$.

3-XOR Problem

Definition (Collision problem)

Given two functions f_0, f_1, find two inputs (x_0, x_1) such that $f_0(x_0) \oplus f_1(x_1) = 0$.

Definition (3-XOR problem)

Given three functions f_0, f_1, f_2, find three inputs (x_0, x_1, x_2) such that $f_0(x_0) \oplus f_1(x_1) \oplus f_2(x_2) = 0$.

Definition (3-XOR problem with lists)

Given three lists L_0, L_1, L_2, find three elements $(e_0, e_1, e_2) \in L_0 \times L_1 \times L_2$ such that $e_0 \oplus e_1 \oplus e_2 = 0$.

Gap of the 3-XOR Problem

Definition (3-XOR problem with lists)

Given three lists L_0, L_1, L_2, find three elements $(e_0, e_1, e_2) \in L_0 \times L_1 \times L_2$ such that $e_0 \oplus e_1 \oplus e_2 = 0$.

Cryptanalysis of n-bit 2EM as a 3-XOR with $2n$-bit elements.

Gap of the 3-XOR Problem

Definition (3-XOR problem with lists)

Given three lists L_0, L_1, L_2, find three elements $(e_0, e_1, e_2) \in L_0 \times L_1 \times L_2$ such that $e_0 \oplus e_1 \oplus e_2 = 0$.

Cryptanalysis of n-bit 2EM as a 3-XOR with $2n$-bit elements.

Solving Random 3-XOR with $2n$-bit elements

Requires $|L_0| \cdot |L_1| \cdot |L_2| = 2^{2n}$ so at least one list of size $2^{2n/3}$.
$|L_0| = |L_1| = |L_2| = 2^{2n/3}$ is enough: compute sum of all triples to find a solution.

So we have a proof and Information Theoretical attack in $2^{2n/3}$.
However best algorithms run in time $T = \mathcal{O}(2^n/n)$...

Gap of the 3-XOR Problem

Definition (3-XOR problem with lists)

Given three lists L_0, L_1, L_2, find three elements $(e_0, e_1, e_2) \in L_0 \times L_1 \times L_2$ such that $e_0 \oplus e_1 \oplus e_2 = 0$.

Cryptanalysis of n-bit 2EM as a 3-XOR with $2n$-bit elements.

Solving Random 3-XOR with $2n$-bit elements

Requires $|L_0| \cdot |L_1| \cdot |L_2| = 2^{2n}$ so at least one list of size $2^{2n/3}$.
$|L_0| = |L_1| = |L_2| = 2^{2n/3}$ is enough: compute sum of all triples to find a solution.

So we have a proof and Information Theoretical attack in $2^{2n/3}$.
However best algorithms run in time $T = \mathcal{O}(2^n/n)$...

$\Longrightarrow$ We found the same gap... again !

Our Strategy

3-XOR solving

Two main techniques:
Multicollision based [Nikolic&Sasaki15] and Linear algebra based [Joux09].
Roughly same asymptotic time complexity.

Our Strategy

3-XOR solving

Two main techniques:
Multicollision based [Nikolic&Sasaki15] and Linear algebra based [Joux09].
Roughly same asymptotic time complexity.

2EM cryptanalysis

Except for one, [DDKS16], all previous cryptanalysis use multicollision based techniques.

Exhibiting the link to 3-XOR allows us to deeply explore linear algebra based techniques for cryptanalysis.
Benefits : Reduced online complexity AND memory both arguably costlier than time.

2-Round Even-Mansour: Results

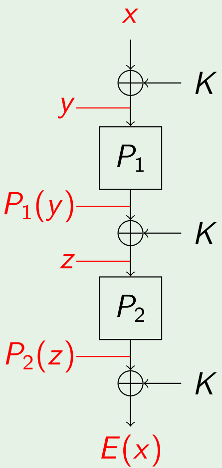
Ref	Data		Queries	Time	Memory	Param.
[NWW13]	$2^n \ln n / n$	KP	$2^n \ln n / n$	$2^n \ln n / n$	$2^n \ln n / n$	
[DDKS13]	$2^{\lambda n}$	KP	$2^n \ln n / n$	$2^n \ln n / n$	$2^n \ln n / n$	
[DDKS16]	$2^n / \lambda n$	CP	$2^n / \lambda n$	$2^n / \lambda n$	$2^{\lambda n}$	$0 < \lambda < \frac{1}{3}$
[IsoShi17]	$2^n \ln n / n$	CP	$2^n \ln n / n$	$2^n \ln n / n$	$2^n \ln n / n$	
	$2^{\lambda n}$	CP	$2^n \ln n / n$	$2^n \ln n / n$	$2^n \ln n / n$	
	$2^n \beta / n$	CP	$2^n / 2^\beta$	$2^n \beta / n$	$2^n / 2^\beta$	$\log n \leq \beta < n$
This Work	n	KP	$2^n / \sqrt{n}$	$2^n / \sqrt{n}$	$2^n / \sqrt{n}$	
This Work	2^d	KP	$2^{n-d/2}$	$2^n / n$	$2^{n-d/2}$	$0 < d < n$
This Work	2^d	KP	$2^{n-d/2}$	$2^n \ln^2 n / n^2$	$2^{n-d/2}$	$0 < d < n$
This Work	λn	KP	$2^n / \lambda n$	$2^n / \lambda n$	$2^{\lambda n}$	$0 < \lambda < 1$

red means $\tilde{\Theta}(2^n)$

First attack on 2EM

1. $L_0 \ni \quad x \quad || \quad x \oplus E(x)$
2. $L_1 \ni \quad y \oplus P_1(y) \quad || \quad y$
3. $L_2 \ni \quad z \quad || \quad P_2(z)$
4. Solve the 3-XOR over L_0, L_1, L_2.
5. Guess $K = x \oplus y$ for the solution found.

2EM



Joux's Technique

$$L_0 = \overbrace{\begin{array}{ccccccccccccccccccccc}
1 & \cdot & 1 & \cdot & 1 & 1 & \cdot & 1 & 1 & \cdot & 1 & \cdot & \cdot & 1 & \cdot & 1 & \cdot & 1 & 1 & 1 & 1 \\
\cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & 1 \\
1 & 1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 & 1 & 1 \\
1 & \cdot & 1 & \cdot & 1 & \cdot & \cdot & 1 & \cdot & 1 & 1 & 1 & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 \\
1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 & 1 & \cdot & \cdot \\
1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 \\
\cdot & 1 & \cdot & 1 & 1 & \cdot & \cdot & 1 & 1 & \cdot & 1 & \cdot & 1 & \cdot & 1 & 1 & \cdot & \cdot & 1 & 1 & 1 \\
1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & 1 & 1 & 1 & 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & 1 \\
1 & \cdot & 1 & 1 & \cdot & 1 & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & 1 & 1 \\
\cdot & \cdot & 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot
\end{array}}^{2n \text{ bits}} \; n \text{ words}$$

Joux's Technique

$$L_0 \cdot M = \overbrace{\begin{array}{ccccccccccccccccccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{array}}^{2n \text{ bits}} \left.\vphantom{\begin{array}{c}1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\end{array}}\right\} n \text{ words}$$

$$e_0 \oplus e_1 \oplus e_2 = 0 \iff e_0 \cdot M \oplus e_1 \cdot M \oplus e_2 \cdot M = 0$$

3-XOR with $L_0, L_1, L_2 \iff$ 3-XOR with $L_0 \cdot M, L_1 \cdot M, L_2 \cdot M$

Joux's Technique

1. Compute M s.t. $L_0 \cdot M = 0_n || I_n$;
2. $L_1' = L_1 \cdot M$;
3. $L_2' = L_2 \cdot M$;
4. Look for partial n-bit collisions between L_1' and L_2';
5. Check if Solution.

Complexity

$|L_0| = n$
$|L_1| = |L_2| = \frac{2^n}{\sqrt{n}}$ $\implies |L_0| \cdot |L_1| \cdot |L_2| = 2^{2n} \checkmark$

$\mathcal{O}(\frac{2^n}{\sqrt{n}})$ memory and computations.

First attack on 2EM

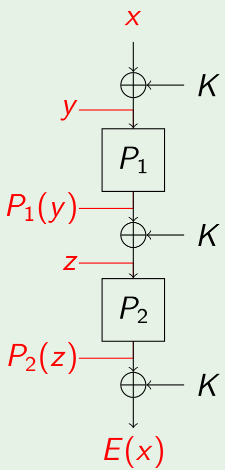
1. $L_0 \ni \quad x \quad || \quad x \oplus E(x)$
2. $L_1 \ni \quad y \oplus P_1(y) \quad || \quad y$
3. $L_2 \ni \quad z \quad || \quad P_2(z)$
4. Solve the 3-XOR over L_0, L_1, L_2.
5. Guess $K = x \oplus y$ for the solution found.

Complexity using Joux's technique $w = 2n$

$D = n$ online queries (Known Plaintext)
$Q = \frac{2^n}{\sqrt{n}}$ offline queries

$\mathcal{O}(\frac{2^n}{\sqrt{n}})$ memory and computations.

2EM

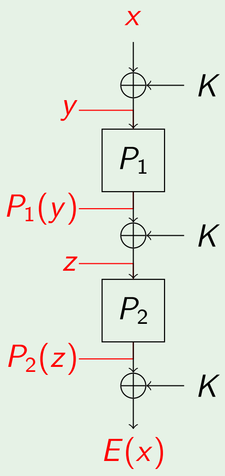


Easy Clamping

We are NOT in the random 3-XOR case.

1. $L_0 \ni \quad x \quad || \quad x \oplus E(x)$
2. $L_1 \ni \quad y \oplus P_1(y) \quad || \quad y$
3. $L_2 \ni \quad z \quad || \quad P_2(z)$

2EM

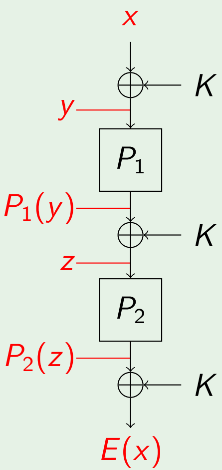


Easy Clamping

We are NOT in the random 3-XOR case.

1. $L_0 \ni \quad x \quad || \quad x \oplus E(x)$
2. $L_1 \ni \quad y \oplus P_1(y) \quad || \quad y$
3. $L_2 \ni \quad P_2^{-1}(z') \quad || \quad z'$

2EM



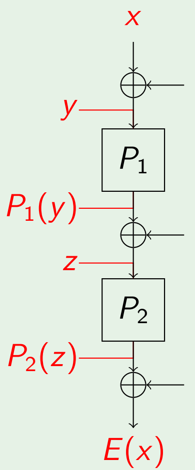
Easy Clamping

We are NOT in the random 3-XOR case.

1. $L_0 \ni \quad x \quad || \quad x \oplus E(x)$
2. $L_1 \ni \quad y \oplus P_1(y) \quad || \quad y$
3. $L_2 \ni \quad P_2^{-1}(z') \quad || \quad z'$

Let $D = 2^d$ thus $Q = 2^{n-d/2}$ $\implies DQ^2 = 2^{2n}$✓
Only compute for y and z' with $d/2$ trailing zeroes.
Only keep $x \oplus E(x)$ with $d/2$ trailing zeroes.

2EM



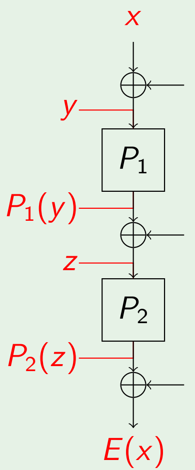
Easy Clamping

We are NOT in the random 3-XOR case.

1. $L_0 \ni \quad x \quad || \quad x \oplus E(x)$
2. $L_1 \ni \quad y \oplus P_1(y) \quad || \quad **|0$
3. $L_2 \ni \quad P_2^{-1}(z') \quad || \quad **|0$

Let $D = 2^d$ thus $Q = 2^{n-d/2}$ $\implies DQ^2 = 2^{2n}$✓
Only compute for y and z' with $d/2$ trailing zeroes.
Only keep $x \oplus E(x)$ with $d/2$ trailing zeroes.

2EM



Easy Clamping

We are NOT in the random 3-XOR case.

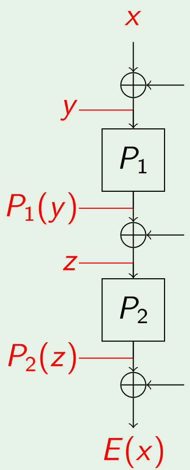
1. $L_0 \ni \quad x \quad || \quad x \oplus E(x)$
2. $L_1 \ni \quad y \oplus P_1(y) \quad || \quad **|0$
3. $L_2 \ni \quad P_2^{-1}(z') \quad || \quad **|0$

Let $D = 2^d$ thus $Q = 2^{n-d/2}$ $\implies DQ^2 = 2^{2n}\checkmark$
Only compute for y and z' with $d/2$ trailing zeroes.
Only keep $x \oplus E(x)$ with $d/2$ trailing zeroes.

3-XOR after clamping

$|L_0| = D/2^{d/2} = 2^{d/2}$
$|L_1| = |L_2| = Q = 2^{n-d/2}$
Reduced lists of $2n - d/2$-bit elements.

2EM



Other 3-XOR algorithms

Generalized 3-XOR algorithm for w-bit elements and $|L_0| \cdot |L_1| \cdot |L_2| = 2^w$:

Wagner's generalized birthday

Combine two lists and look for a collision.

$T = \mathcal{O}\Big((|L_0| \cdot |L_1|) + |L_2|\Big)$ $\qquad M = \mathcal{O}\Big(|L_1| + |L_2|\Big)$

And two more by [Bouillaguet, Delaplace, Fouque. ToSC 2018]:

Other 3-XOR algorithms

Generalized 3-XOR algorithm for w-bit elements and $|L_0| \cdot |L_1| \cdot |L_2| = 2^w$:

Wagner's generalized birthday

Combine two lists and look for a collision.

$T = \mathcal{O}\Big((|L_0| \cdot |L_1|) + |L_2|\Big)$ $\qquad M = \mathcal{O}\Big(|L_1| + |L_2|\Big)$

And two more by [Bouillaguet, Delaplace, Fouque. ToSC 2018]:

Repeat $\mathcal{O}(|L_0|/w)$ times Joux's algorithm.

Realistic 3-XOR algorithm.

$T = \mathcal{O}\Big(|L_0| \cdot (|L_1| + |L_2|)/w\Big)$ $\qquad M = \mathcal{O}\Big(|L_1| + |L_2|\Big)$

Other 3-XOR algorithms

Generalized 3-XOR algorithm for w-bit elements and $|L_0| \cdot |L_1| \cdot |L_2| = 2^w$:

Wagner's generalized birthday

Combine two lists and look for a collision.

$T = \mathcal{O}\Big((|L_0| \cdot |L_1|) + |L_2|\Big)$ $M = \mathcal{O}\Big(|L_1| + |L_2|\Big)$

And two more by [Bouillaguet, Delaplace, Fouque. ToSC 2018]:

Repeat $\mathcal{O}(|L_0|/w)$ times Joux's algorithm.

Realistic 3-XOR algorithm.

$T = \mathcal{O}\Big(|L_0| \cdot (|L_1| + |L_2|)/w\Big)$ $M = \mathcal{O}\Big(|L_1| + |L_2|\Big)$

Revisited Baran-Demaine-Pătrașcu 3-SUM algorithm

Best known asymptotic complexity but impractical for realistic w.

$T = \mathcal{O}\Big((|L_0| \cdot |L_1| + |L_2|) \cdot \ln^2(w)/w^2\Big)$ $M = \mathcal{O}\Big(|L_1| + |L_2|\Big)$

Other 3-XOR algorithms

Generalized 3-XOR algorithm for w-bit elements and $|L_0| \cdot |L_1| \cdot |L_2| = 2^w$:

Wagner's generalized birthday

Combine two lists and look for a collision.

$T = \mathcal{O}\left(2^n\right)$ $\qquad M = \mathcal{O}\left(2^{n-d/2}\right)$

And two more by [Bouillaguet, Delaplace, Fouque. ToSC 2018]:

Repeat $\mathcal{O}(|L_0|/w)$ times Joux's algorithm.

Realistic 3-XOR algorithm.

$T = \mathcal{O}\left(2^n/n\right)$ $\qquad M = \mathcal{O}\left(2^{n-d/2}\right)$

Revisited Baran-Demaine-Pătrașcu 3-SUM algorithm

Best known asymptotic complexity but impractical for realistic w.

$T = \mathcal{O}\left(2^n \cdot \ln^2(n)/n^2\right)$ $\qquad M = \mathcal{O}\left(2^{n-d/2}\right)$

2-Round Even-Mansour: Results

Strategy	Data		Queries	Time	Memory	Param.
Joux's technique	n	KP	$2^n/\sqrt{n}$	$2^n/\sqrt{n}$	$2^n/\sqrt{n}$	
Clamping + BDF algo	2^d	KP	$2^{n-d/2}$	$2^n/n$	$2^{n-d/2}$	$0 < d < n$
Clamping + BDP algo	2^d	KP	$2^{n-d/2}$	$2^n \ln^2 n/n^2$	$2^{n-d/2}$	$0 < d < n$
Low-Data	λn	KP	$2^n/\lambda n$	$2^n/\lambda n$	$2^{\lambda n}$	$0 < \lambda < 1$

red means $\tilde{\Theta}(2^n)$

Joux's Technique

$$L_0 \cdot M = \begin{array}{c}2n \text{ bits}\\ \overleftrightarrow{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}}\\ \begin{matrix}
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1
\end{matrix}\end{array} \updownarrow n \text{ words}$$

$$e_0 \oplus e_1 \oplus e_2 = 0 \iff e_0 \cdot M \oplus e_1 \cdot M \oplus e_2 \cdot M = 0$$

$$\text{3-XOR with } L_0, L_1, L_2 \iff \text{3-XOR with } L_0 \cdot M, L_1 \cdot M, L_2 \cdot M$$

Joux's Technique... but smaller

$$L_0^1 || L_0^2 = \overbrace{\begin{matrix}
\cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 \\
\cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot \\
1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & 1 & 1 & 1 & 1 & 1 & \cdot & 1 & 1 & 1 \\
1 & \cdot & 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\
1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot
\end{matrix}}^{2n \text{ bits}} \quad \lambda n \text{ words}$$

Joux's Technique... but smaller

$$
L_0^1||(L_0^2 \cdot M_s) = \overbrace{
\underbrace{\begin{matrix}
\cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & \cdot \\
1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 \\
\cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 \\
1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot \\
1 & \cdot & 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & \cdot \\
1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot & 1 & 1
\end{matrix}}_{n}
\;
\underbrace{\begin{matrix}
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot
\end{matrix}}_{(1-\lambda)n}
\;
\underbrace{\begin{matrix}
1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{matrix}}_{\lambda n}
}^{2n \text{ bits}}
\Bigg\} \lambda n \text{ words}
$$

Joux's Technique... but smaller

$$L_0^1 || (L_0^2 \cdot M_s) = \overbrace{\underbrace{\begin{array}{cccccccccc} \cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 \\ 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot \\ 1 & \cdot & 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & \cdot \\ 1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot & 1 & 1 \end{array}}_{n} \underbrace{\begin{array}{ccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array}}_{(1-\lambda)n} \underbrace{\begin{array}{cccccc} 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{array}}_{\lambda n}}^{2n \text{ bits}} \; \lambda n \text{ words}$$

$$L_1 \ni \quad y \oplus P_1(y) \quad || \quad y$$

$$L_2 \ni \quad P_2^{-1}(z') \quad || \quad z'$$

Joux's Technique... but smaller

$2n$ bits

$$L_0^1||(L_0^2 \cdot M_s) = \begin{array}{cccccccccc|ccccc|cccccc}
\cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
1 & \cdot & 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{array} \quad \lambda n \text{ words}$$

$$\qquad\qquad n \qquad\qquad\qquad (1-\lambda)n \qquad\qquad \lambda n$$

$$L_1^1||(L_1^2 \cdot M_s) \ni \quad y \oplus P_1(y) \quad || \quad y \cdot M_s$$

$$L_2^1||(L_2^2 \cdot M_s) \ni \quad P_2^{-1}(z') \quad || \quad z' \cdot M_s$$

Joux's Technique... but smaller

$$
L_0^1 || (L_0^2 \cdot M_s) = \overbrace{\begin{array}{ccccccccccccccccccccc}
\cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
1 & \cdot & 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{array}}^{2n \text{ bits}} \; \lambda n \text{ words}
$$

$$
n \qquad (1-\lambda)n \qquad \lambda n
$$

$$
L_1^1 || (L_1^2 \cdot M_s) \ni \quad y \oplus P_1(y) \quad || \quad \alpha \quad | \quad * * *
$$

$$
L_2^1 || (L_2^2 \cdot M_s) \ni \quad P_2^{-1}(z') \quad || \quad \alpha \quad | \quad * * *
$$

Joux's Technique... but smaller

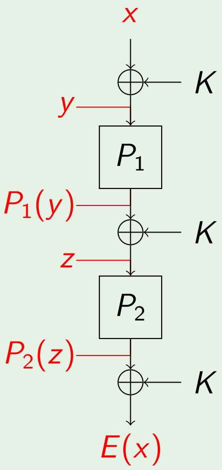
$$
(L_0^1 || L_0^2) \cdot M = \overbrace{\underbrace{\begin{matrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{matrix}}_{n \;|\; (1-\lambda)n \;|\; \lambda n}}^{2n \text{ bits}} \left.\vphantom{\begin{matrix}1\\1\\1\\1\\1\\1\end{matrix}}\right\updownarrow \lambda n \text{ words}
$$

	n		$(1-\lambda)n$		λn
$L_1^1 \| (L_1^2 \cdot M_s) \ni$	$y \oplus P_1(y)$	$\|\|$	α	$\|$	$* * *$
$L_2^1 \| (L_2^2 \cdot M_s) \ni$	$P_2^{-1}(z')$	$\|\|$	α	$\|$	$* * *$

Low-Data Attack on 2EM

Collision over $(1-\lambda)n$ bits for free.
L_1 and L_2 contain $2^{\lambda n}$ elements and reused for different α.

2EM



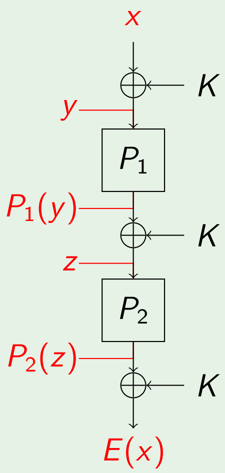
Low-Data Attack on 2EM

Collision over $(1-\lambda)n$ bits for free.
L_1 and L_2 contain $2^{\lambda n}$ elements and reused for different α.

Complexity

Data $D = \lambda n$.
Memory $\mathcal{O}(2^{\lambda n})$.
Time $T = Q = \mathcal{O}(\frac{2^n}{\lambda n})$.

2EM

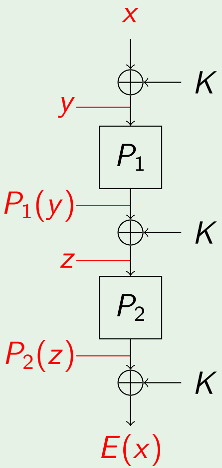


Some Take-aways

Clamping + algo

After easy clamping we can use a generic 3-XOR algorithm.
Faster 3-XOR solver $\implies$ Faster 2EM cryptanalysis!

2EM



Some Take-aways

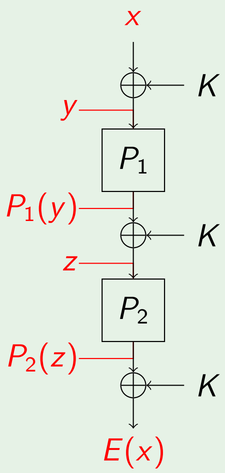
Clamping + algo

After easy clamping we can use a generic 3-XOR algorithm.
Faster 3-XOR solver $\implies$ Faster 2EM cryptanalysis!

Linear algebra vs Multicollision

Roughly as much computations.
But less memory.

2EM



Some Take-aways

Clamping + algo

After easy clamping we can use a generic 3-XOR algorithm.
Faster 3-XOR solver $\implies$ Faster 2EM cryptanalysis!

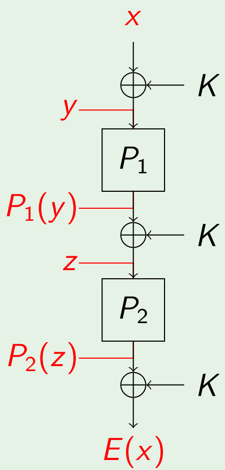
Linear algebra vs Multicollision

Roughly as much computations.
But less memory.

Low-Data Attack

Uses $D = \lambda n$ and $T = 2^n/(\lambda n)$.
$\implies DT = 2^n$
Matches the 1EM proof $DT \leq 2^n$ for $0 < \lambda \leq 1 - \frac{\ln(n \ln 2)}{n \ln 2} + o(1)$.

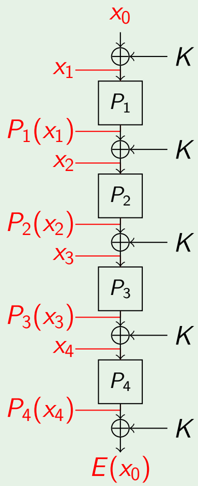
2EM



Generalization of the Reduction

We've shown 2EM as a 3-XOR with 2n-bit elements and...

4EM



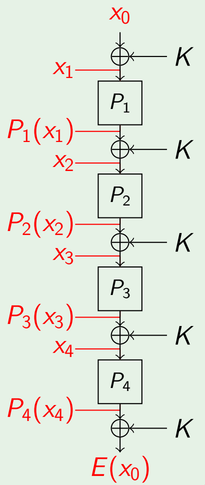
Generalization of the Reduction

We've shown 2EM as a 3-XOR with $2n$-bit elements and...

Lists for 4EM cryptanalysis using the 5-XOR problem.

$L_0 \ni$	x_0	.	.	$E(x_0)$
$L_1 \ni$	$x_1 \oplus P_1(x_1)$	$P_1(x_1)$	.	.
$L_2 \ni$	x_2	$x_2 \oplus P_2(x_2)$	$P_2(x_2)$	.
$L_3 \ni$	.	x_3	$x_3 \oplus P_3(x_3)$	$P_3(x_3)$
$L_4 \ni$	.	.	x_4	$x_4 \oplus P_4(x_4)$

4EM



Generalization of the Reduction

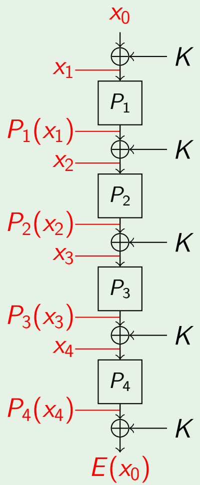
We've shown 2EM as a 3-XOR with $2n$-bit elements and...

Lists for 4EM cryptanalysis using the 5-XOR problem.

$L_0 \ni$	x_0	.	.	$E(x_0)$
$L_1 \ni$	$x_1 \oplus P_1(x_1)$	$P_1(x_1)$	.	.
$L_2 \ni$	x_2	$x_2 \oplus P_2(x_2)$	$P_2(x_2)$	.
$L_3 \ni$	.	x_3	$x_3 \oplus P_3(x_3)$	$P_3(x_3)$
$L_4 \ni$	.	.	x_4	$x_4 \oplus P_4(x_4)$

rEM cryptanalysis as a special $(r+1)$-XOR with rn-bit elements.
Can we use this to improve cryptanalysis of rEM with $r \geq 3$?

4EM



2-Round Even-Mansour: Results

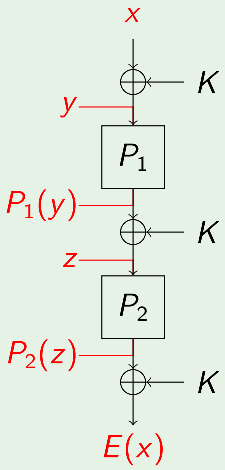
Strategy	Data		Queries	Time	Memory	Param.
Joux's technique	n	KP	$2^n/\sqrt{n}$	$2^n/\sqrt{n}$	$2^n/\sqrt{n}$	
Clamping + BDF algo	2^d	KP	$2^{n-d/2}$	$2^n/n$	$2^{n-d/2}$	$0 < d < n$
Clamping + BDP algo	2^d	KP	$2^{n-d/2}$	$2^n \ln^2 n/n^2$	$2^{n-d/2}$	$0 < d < n$
Low-Data	λn	KP	$2^n/\lambda n$	$2^n/\lambda n$	$2^{\lambda n}$	$0 < \lambda < 1$

- Link between 2EM cryptanalysis and the 3-XOR Problem.
- Explore existing and new linear algebra techniques.
- Significantly reduce online data and memory usage (previous bottleneck).

Low-Data Attack on 2EM

1. Collect λn plaintext/ciphertext pairs for L_0 and compute M_s.
2. Pick a new $(1 - \lambda n)$-bit value α:
   2.1 For all λn-bit value u: let $y = z' = (\alpha \| u) \cdot M_s^{-1}$ and fill L_1 and L_2.
   2.2 Solve the 3-XOR over L_0, L_1, L_2 using Joux's technique. (Only an $(n + \lambda n)$-bit collision)
   2.3 Clear L_1 and L_2. Loop if no solution.
3. Guess $K = x \oplus y$ for the solution found.

2EM



Low-Data Attack on 2EM

1. Collect λn plaintext/ciphertext pairs for L_0 and compute M_s.
2. Pick a new $(1-\lambda n)$-bit value α:
   2.1 For all λn-bit value u: let $y = z' = (\alpha \| u) \cdot M_s^{-1}$ and fill L_1 and L_2.
   2.2 Solve the 3-XOR over L_0, L_1, L_2 using Joux's technique. (Only an $(n+\lambda n)$-bit collision)
   2.3 Clear L_1 and L_2. Loop if no solution.
3. Guess $K = x \oplus y$ for the solution found.

Complexity of Low-Data Attack

Each loop pr. of success: $\lambda n 2^{2\lambda n}/2^{(n+\lambda n)} = \lambda n 2^{\lambda n - n}$.
Each loop uses $2^{\lambda n}$ computations.
$D = \lambda n$.
$T = Q = \mathcal{O}(\frac{2^n}{\lambda n})$.
$\mathcal{O}(2^{\lambda n})$ memory.

2EM