Cryptanalysis of FIDES

Itai Dinur[1] Jérémy Jean[1,2]

[1]École Normale Supérieure, France

[2]Nanyang Technological University, Singapore

FSE 2014 – March 3, 2014

Authenticated Encryption (AE)

Motivations

- Crypto is not *only* about encryption
- Integrity and authenticity are often required
- Existing solutions (modes, MAC)
- Few dedicated ciphers
- Recent focus on this topic with the CAESAR competition

Regular cipher	AE	AEAD
$(\mathcal{M}, \mathcal{K}) \longrightarrow \mathcal{C}$	$(\mathcal{M}, \mathcal{K}) \longrightarrow (\mathcal{C}, \mathcal{T})$	$(\mathcal{M}, \mathcal{K}, \mathcal{A}) \longrightarrow (\mathcal{C}, \mathcal{T}, \mathcal{A})$

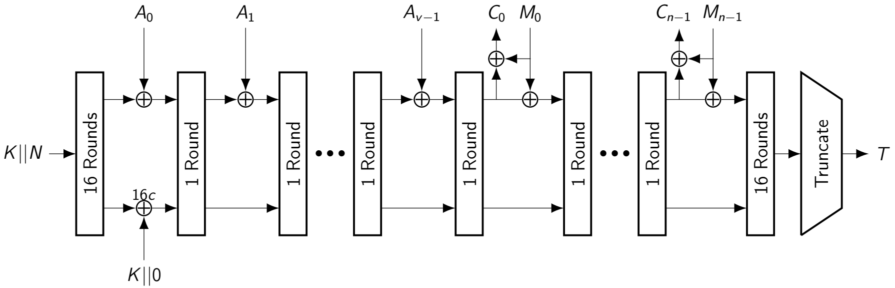
$\mathcal{M}$: plaintext
$\mathcal{C}$: ciphertext
$\mathcal{K}$: key

$\mathcal{T}$: authentication tag
$\mathcal{A}$: optional associated data

Description of FIDES (1/2)

FIDES

- Designed by Bilgin et al. and published at CHES 2013
- Nonce-based lightweight authenticated cipher (N)
- Key sizes: 80 and 96 bits (K)
- Handle optional associated data (A)
- Leak-extraction structure similar to the duplex sponge construction
- Permutation: application of an unkeyed AES round



Description of FIDES (2/2)

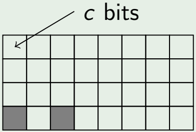
Internal state:

- Internal state of $4 \times 8 \times c$ bits
- Nibble size c:
  - $c = 5$ for FIDES−80
  - $c = 6$ for FIDES−96

One Round of the Internal Permutation:

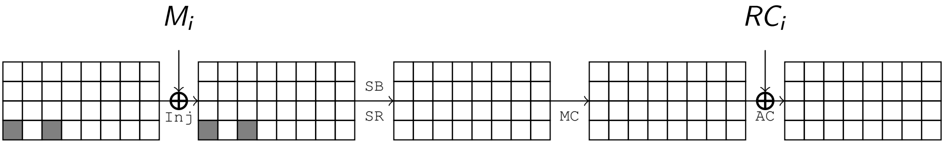
- Extract $2c$-bit mask
- $2c$-bit message injection
- AES-like operations: SB, SR, MC, AC.
- Suboptimal diffusion matrix (non MDS)

Internal state



Diffusion Matrix

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



Leakage and Security Claims

Leakage

- The same positions are used to leak and inject nibbles
- $2c$ out of $32c$ bits are leaked before each round

Security Claims

- Nonce-respecting adversary assumption
- Attack scenarios: state recovery, key recovery and forgery
- FIDES advertises 16c-bit security against all scenarios

Our Attack

- State recovery can be done in 2^{15c} operations
- We can forge *any* message after a state recovery

Similar designs

FIDES is reminiscent of other AES-based design using leak-extraction.

LEX [Bir06]

- 128-bit key stream cipher
- 4/16 leaked nibbles per round
- No injection (stream cipher)

ALE [BMR+13]

- 128-bit AE cipher
- 4/16 leaked nibbles per round
- Inject 16 nibbles every 4 rounds

Alpha-MAC [DR05]

- 128-bit MAC
- 4 nibbles injected per round
- No extraction

ASC-1 [JK11]

- 128-bit AE cipher
- 4/16 leaked nibbles per round
- Inject 16 nibbles every 4 rounds
- Whitening key before leakage

Similar designs

FIDES is reminiscent of other AES-based design using leak-extraction.

LEX [Bir06]

- 128-bit key stream cipher
- 4/16 leaked nibbles per round
- No injection (stream cipher)

Broken [DK13, BDF11]

ALE [BMR+13]

- 128-bit AE cipher
- 4/16 leaked nibbles per round
- Inject 16 nibbles every 4 rounds

Broken [KR13]

Alpha-MAC [DR05]

- 128-bit MAC
- 4 nibbles injected per round
- No extraction

Broken [YWJ+09, BDF11]

ASC-1 [JK11]

- 128-bit AE cipher
- 4/16 leaked nibbles per round
- Inject 16 nibbles every 4 rounds
- Whitening key before leakage

Results on FIDES

Results

Cipher	Data	Time	Memory	Generic	Ref
FIDES-80	1 KP	2^{75}	2^{15}	2^{80}	This paper
	2^{64} KP	2^{73}	2^{64}	2^{80}	Long version
FIDES-96	1 KP	2^{90}	2^{18}	2^{96}	This paper
	2^{77} KP	2^{88}	2^{77}	2^{96}	Long version

Notes:

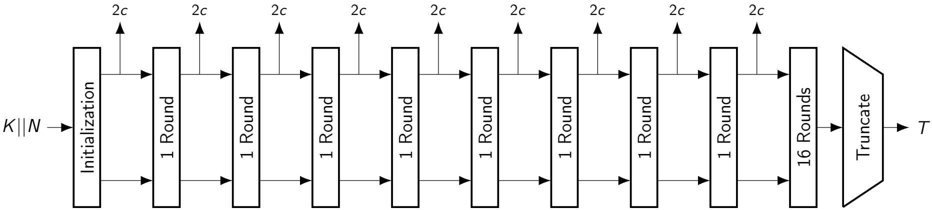
- Guess-and-determine attacks
- Recover the internal state
- Allow to forge arbitrary messages

Preliminaries (1/2)

How many leaked nibbles are needed to recover the state faster than exhaustive search?

Information theoretically speaking:

- The state consists of 32 nibbles
- Known-plaintext scenario
- 15 rounds would leak a total $(15 + 1) \times 2 = 32$ state nibbles
- Uniquely determine the state
- But analyzing 15 consecutive AES-like rounds is difficult



Preliminaries (2/2)

With $n \in [0, 14]$ rounds:

- Reduce the analysis to n consecutive AES-like rounds
- A total of $(n + 1) \times 2$ state nibbles are leaked
- Unicity of the state no longer true: about $2^{(32-2n-2)\times c}$ different initial states would leak the same sequence
- Goal: Generating all of them in less than 2^{16c} computations
- $32 - 2n - 2 < 16 \Longrightarrow n \geq 8$.

Preliminaries (2/2)

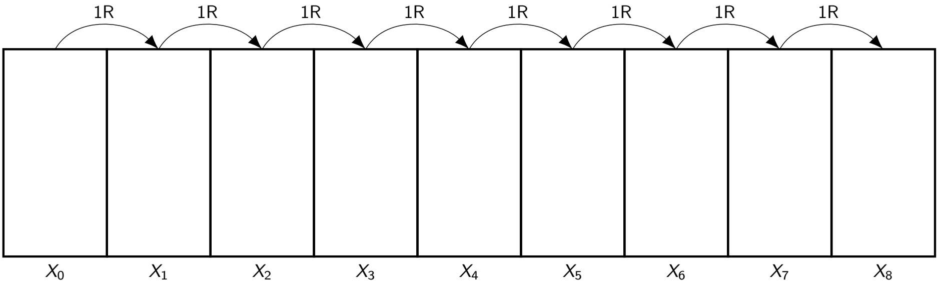
With $n \in [0, 14]$ rounds:

- Reduce the analysis to n consecutive AES-like rounds
- A total of $(n + 1) \times 2$ state nibbles are leaked
- Unicity of the state no longer true: about $2^{(32-2n-2)\times c}$ different initial states would leak the same sequence
- Goal: Generating all of them in less than 2^{16c} computations
- $32 - 2n - 2 < 16 \Longrightarrow n \geq 8$.

Our Attack

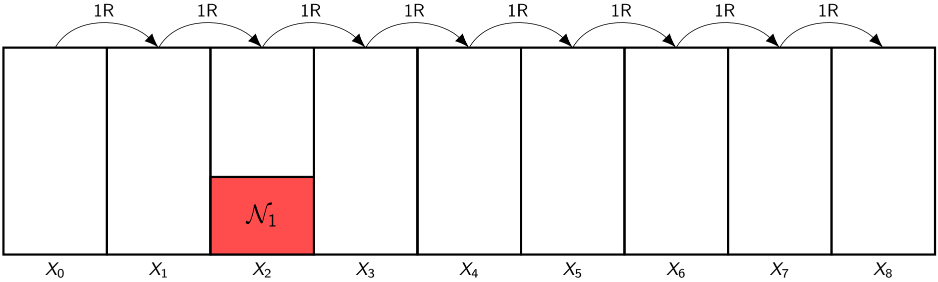
- We use the knowledge of 18 leaked nibbles, in 9 consecutive states linked by $n = 8$ rounds (in fact, only 17 nibbles)
- Data: less than 16 bytes of a single known plaintext
- Time: about 2^{15c} computations to enumerate the $2^{(32-17)c} = 2^{15c}$ state candidates
- Check: additional leaked bytes, or authentication tag T.

High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

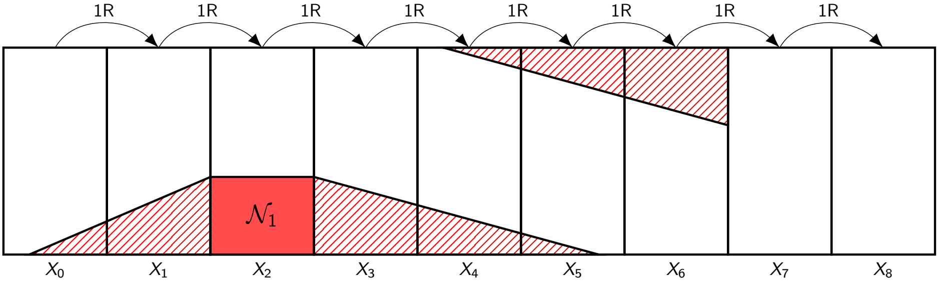
High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

1. Guess the 12 nibbles in the set $\mathcal{N}_1$

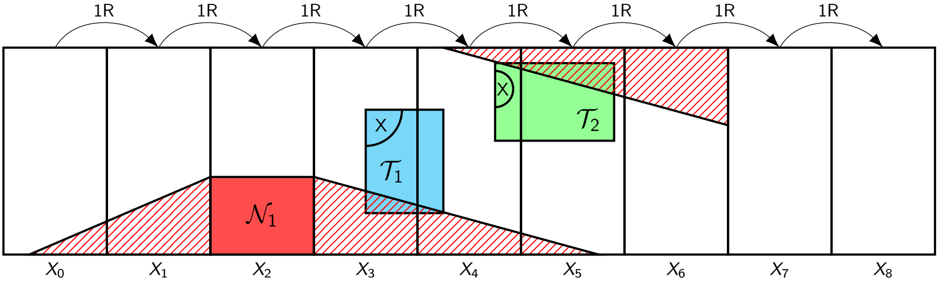
High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

1. Guess the 12 nibbles in the set $\mathcal{N}_1$
2. Determine other nibble values ($\mathcal{N}_1'$)

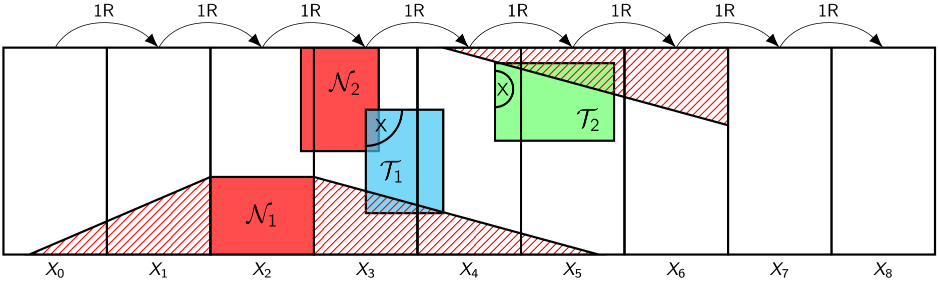
High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

1. Guess the 12 nibbles in the set $\mathcal{N}_1$
2. Determine other nibble values ($\mathcal{N}_1'$)
3. Construct two tables $\mathcal{T}_1$ and $\mathcal{T}_2$ (independently)

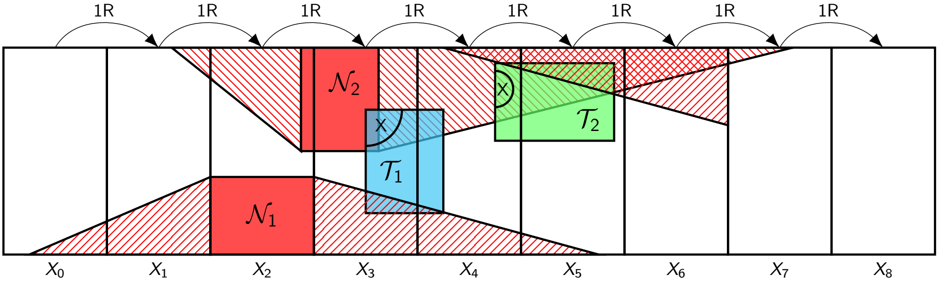
High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

1. Guess the 12 nibbles in the set $\mathcal{N}_1$
2. Determine other nibble values ($\mathcal{N}_1'$)
3. Construct two tables $\mathcal{T}_1$ and $\mathcal{T}_2$ (independently)
4. Guess the 3 nibbles in the set $\mathcal{N}_2$

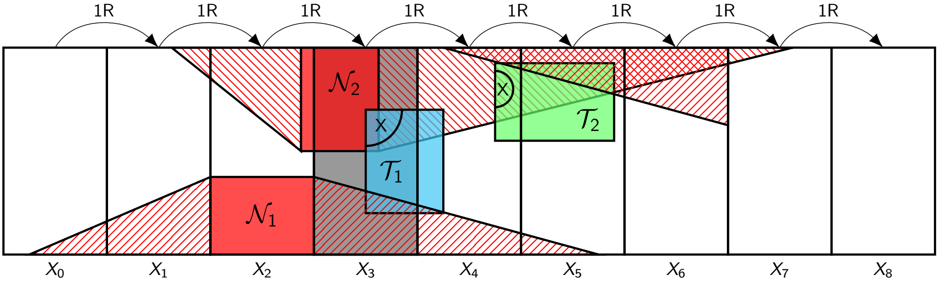
High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

1. Guess the 12 nibbles in the set $\mathcal{N}_1$
2. Determine other nibble values ($\mathcal{N}_1'$)
3. Construct two tables $\mathcal{T}_1$ and $\mathcal{T}_2$ (independently)
4. Guess the 3 nibbles in the set $\mathcal{N}_2$
5. Determine new nibble values ($\mathcal{N}_2'$)

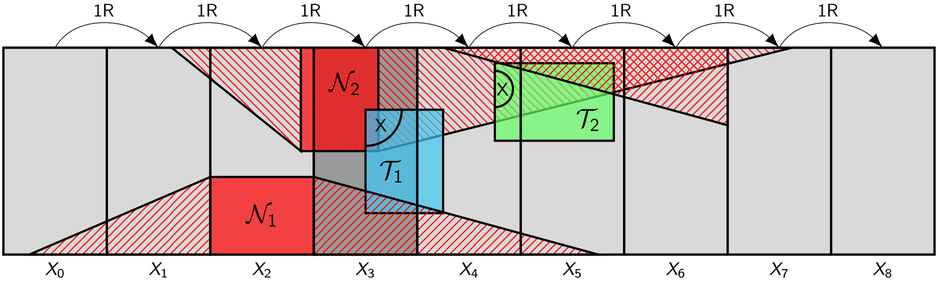
High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

1. Guess the 12 nibbles in the set $\mathcal{N}_1$
2. Determine other nibble values ($\mathcal{N}_1'$)
3. Construct two tables $\mathcal{T}_1$ and $\mathcal{T}_2$ (independently)
4. Guess the 3 nibbles in the set $\mathcal{N}_2$
5. Determine new nibble values ($\mathcal{N}_2'$)
6. Use the tables $\mathcal{T}_1$ and $\mathcal{T}_2$ to fully recover a middle state

High-Level Overview of the State-Recovery Attack



Steps of the Guess-and-determine Procedure

1. Guess the 12 nibbles in the set $\mathcal{N}_1$
2. Determine other nibble values ($\mathcal{N}_1'$)
3. Construct two tables $\mathcal{T}_1$ and $\mathcal{T}_2$ (independently)
4. Guess the 3 nibbles in the set $\mathcal{N}_2$
5. Determine new nibble values ($\mathcal{N}_2'$)
6. Use the tables $\mathcal{T}_1$ and $\mathcal{T}_2$ to fully recover a middle state

Main Property

The guess-and-determine algorithm relies on the MC matrix which has a branching number of 4 (non MDS, AES: 5).

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Let $\mathbf{x} = [x_0, x_1, x_2, x_3]$ and $\mathbf{y} = [y_0, y_1, y_2, y_3]$.
There are linear dependencies between 4 nibbles of $\mathbf{x}$ and $\mathbf{y} = \mathbf{M}\mathbf{x}$.

Property 1

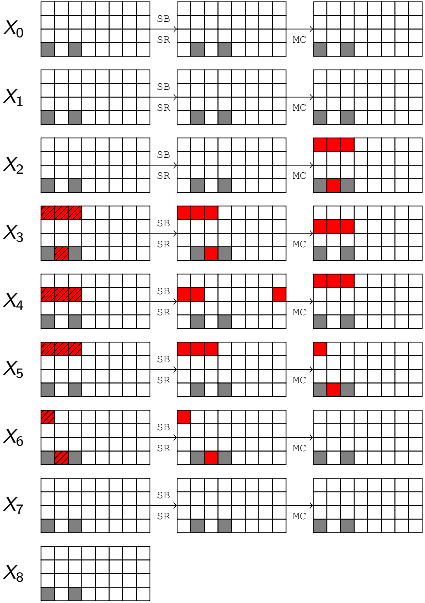
For all $i, j \in \{0, 1, 2, 3\}$ such that $i \neq j$: $x_i \oplus x_j = y_i \oplus y_j$.

Property 2

For all $i \in \{0, 1, 2, 3\}$: $x_{i+3} = y_i \oplus x_{i+1} \oplus x_{i+2}$ (addition mod 4)

$y_{i+3} = x_i \oplus y_{i+1} \oplus y_{i+2}$.

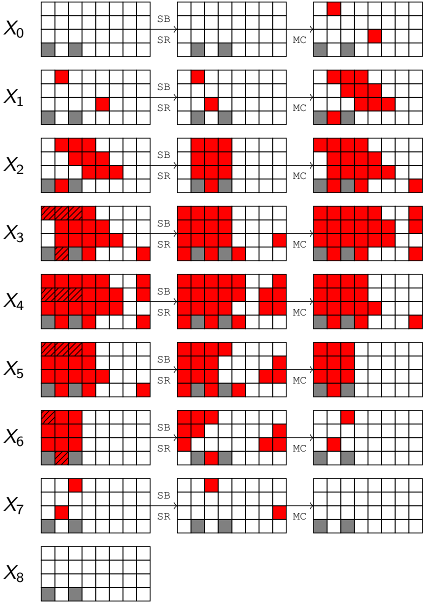
Step 1



$\mathcal{N}_1$

$X_3[0,0], X_3[0,1], X_3[0,2], X_3[3,1],$
$X_4[1,0], X_4[1,1], X_4[1,2],$
$X_5[0,0], X_5[0,1], X_5[0,2],$
$X_6[0,0], X_6[3,1]$

Step 1

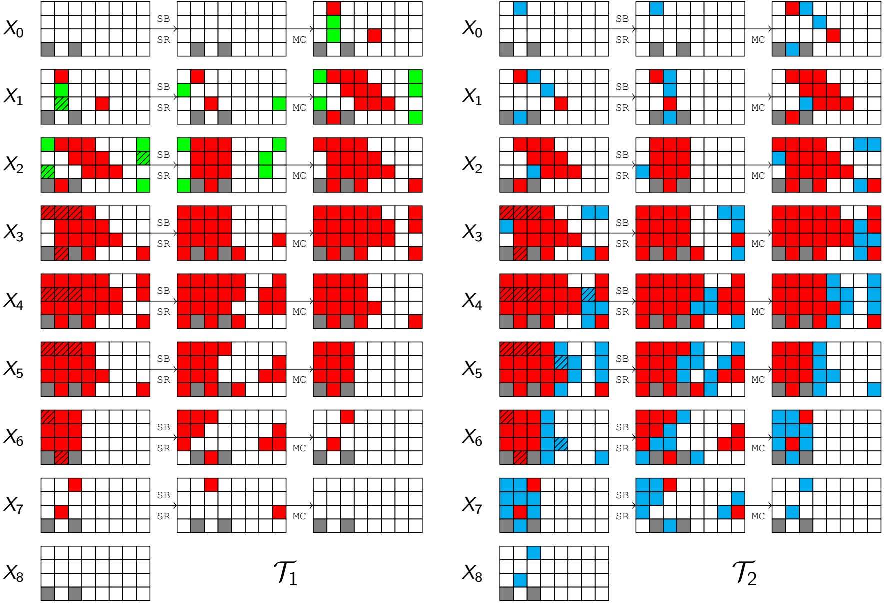


Propagate($\mathcal{N}_1$) $\Longrightarrow$ $\mathcal{N}_1'$

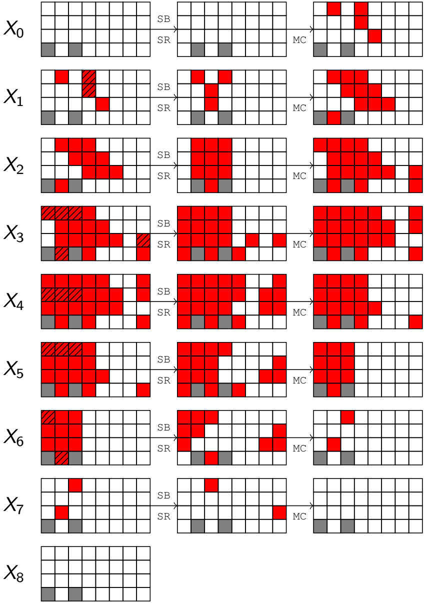
$\mathcal{N}_1'$

$X_1[0,1]$ $X_1[2,4]$ $X_2[0,1]$ $X_2[0,2]$ $X_2[0,3]$
$X_2[1,2]$ $X_2[1,3]$ $X_2[1,4]$ $X_2[2,3]$ $X_2[2,4]$
$X_2[2,5]$ $X_2[3,1]$ $X_3[0,3]$ $X_3[1,1]$ $X_3[1,2]$
$X_3[1,3]$ $X_3[1,4]$ $X_3[2,1]$ $X_3[2,2]$ $X_3[2,3]$
$X_3[2,4]$ $X_3[2,5]$ $X_3[3,3]$ $X_3[3,7]$ $X_4[0,0]$
$X_4[0,1]$ $X_4[0,2]$ $X_4[0,3]$ $X_4[0,4]$ $X_4[0,7]$
$X_4[1,3]$ $X_4[1,4]$ $X_4[1,5]$ $X_4[1,7]$ $X_4[2,0]$
$X_4[2,1]$ $X_4[2,2]$ $X_4[2,3]$ $X_4[2,4]$ $X_4[2,5]$
$X_4[3,1]$ $X_4[3,3]$ $X_4[3,7]$ $X_5[0,3]$ $X_5[1,0]$
$X_5[1,1]$ $X_5[1,2]$ $X_5[1,3]$ $X_5[2,0]$ $X_5[2,1]$
$X_5[2,2]$ $X_5[2,3]$ $X_5[2,4]$ $X_5[3,1]$ $X_5[3,3]$
$X_5[3,7]$ $X_6[0,1]$ $X_6[0,2]$ $X_6[1,0]$ $X_6[1,1]$
$X_6[1,2]$ $X_6[2,0]$ $X_6[2,1]$ $X_6[2,2]$ $X_7[0,2]$
$X_7[2,1]$

Step 2: Construction of $\mathcal{T}_1$ and $\mathcal{T}_2$



Step 3



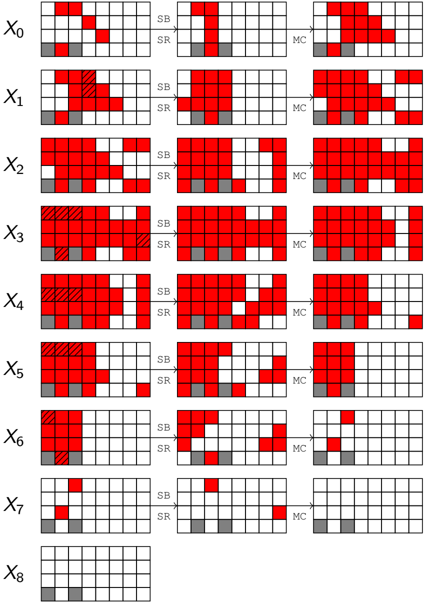
$\mathcal{N}_2$

$X_1[0,3]$,

$X_1[1,3]$,

$X_3[2,7]$

Step 3



Propagate($\mathcal{N}_2$) $\Longrightarrow$ $\mathcal{N}_2'$

$\mathcal{N}_2'$

$X_1[2,3], X_2[2,1], X_1[1,2], X_2[1,1], X_2[2,2], X_3[1,0], X_3[2,0], X_4[2,7], X_3[3,6], X_2[0,0], X_2[3,7], X_3[0,7], X_2[3,6], X_2[0,7], X_3[1,7], X_2[1,0], X_1[2,2], X_1[0,2], X_1[3,1], X_1[1,4], X_1[2,5], X_2[3,3], X_3[0,4], X_3[1,5], X_3[2,6], X_4[3,4], X_3[1,6], X_2[0,6], X_0[0,1], X_0[0,2], X_0[1,3], X_0[2,4], X_0[3,1]$

Final Step: Post-Filtering

The guess-and-determine algorithm:

- Requires $2^{(12+3)c} = 2^{15c}$ computations
- Generates 2^{15c} possible internal states
- We post-filter all those states against extra variables
- we expect only the correct state to remain

Attack Complexity

- **Data:** 17 consecutive leaked nibbles of a KP + additional values
- **Memory:** 2^{3c} elements in tables $\mathcal{T}_1$ and $\mathcal{T}_2$
- **Time:** 2^{15c} computations

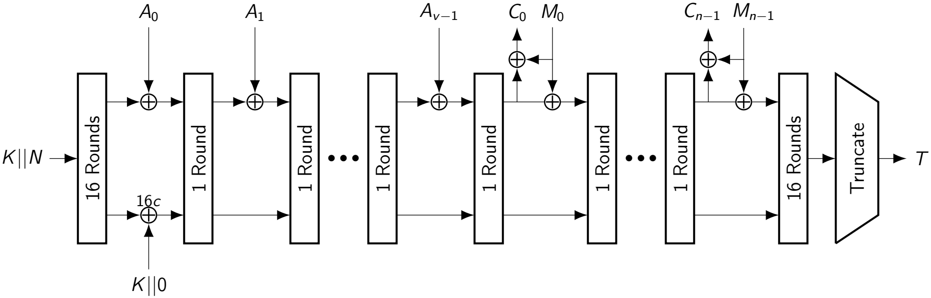
Forgery after the State Recovery

Finalization

The initialization of FIDES does not depend on the message.
The finalization of FIDES does not depend on the key.

Consequently, once the state is recovered:

- we know the state $\mathrm{Init}(K||N)$ after the 16-round initialization
- we can simulate the encryption of any arbitrary message and produce a valid tag



Tradeoffs (Long Version)

Requirements for the tradeoffs

Obtain a t-way collision ($t \geq 2$) on 17 consecutive leaked nibbles.

A t-way collision on the n-bit output of a random map requires about :

$$(t!)^{1/t} \cdot 2^{n(t-1)/t} \text{ evaluations.}$$ [STKT06]

Tradeoffs Points ($n = 17c$)

	FIDES-80 ($c = 5$)		FIDES-96 ($c = 6$)	
t	**Data (KP)**	**Time**	**Data (KP)**	**Time**
2	$2^{42.50}$	$2^{74.00}$	$2^{51.00}$	$2^{89.00}$
3	$2^{56.67}$	$2^{73.42}$	$2^{68.00}$	$2^{88.42}$
4	$2^{63.75}$	$2^{73.00}$	$2^{76.50}$	$2^{88.00}$
5	$2^{68.00}$	$2^{72.68}$	$2^{81.60}$	$2^{87.68}$
6	$2^{70.83}$	$2^{72.42}$	$2^{85.00}$	$2^{87.42}$

KP: known plaintext

Conclusion

Cryptanalysis:

- Guess-and-determine attacks on FIDES AE algorithm
  - State recovery attack
  - Forgery attack
  - Difficult to extend to key-recovery (16-round initialization)
- Very low data complexity: few bytes of a single KP
- Low memory complexity: less than 2^{24} stored elements
- Time complexity:
  - 2^{75} computations for FIDES-80
  - 2^{90} computations for FIDES-96

Possible countermeasures:

- Optimal branching of 5
- Leak (keyed) functions of the state nibbles
- Key-dependent finalization (forgery only)

Conclusion

Cryptanalysis:

- Guess-and-determine attacks on FIDES AE algorithm
  - State recovery attack
  - Forgery attack
  - Difficult to extend to key-recovery (16-round initialization)
- Very low data complexity: few bytes of a single KP
- Low memory complexity: less than 2^{24} stored elements
- Time complexity:
  - 2^{75} computations for FIDES-80
  - 2^{90} computations for FIDES-96

Possible countermeasures:

- Optimal branching of 5
- Leak (keyed) functions of the state nibbles
- Key-dependent finalization (forgery only)

Thank you!

Bibliography I

Charles Bouillaguet, Patrick Derbez, and Pierre-Alain Fouque.
Automatic search of attacks on round-reduced AES and applications.
In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 169–187. Springer, August 2011.

Alex Biryukov.
The design of a stream cipher LEX.
In Eli Biham and Amr M. Youssef, editors, *SAC 2006*, volume 4356 of *LNCS*, pages 67–75. Springer, August 2006.

Andrey Bogdanov, Florian Mendel, Francesco Regazzoni, Vincent Rijmen, and Elmar Tischhauser.
ALE: AES-based lightweight authenticated encryption.
In *FSE*, Lecture Notes in Computer Science, 2013.
to appear.

Orr Dunkelman and Nathan Keller.
Cryptanalysis of the stream cipher LEX.
Des. Codes Cryptography, 67(3):357–373, 2013.

Bibliography II

Joan Daemen and Vincent Rijmen.
A new MAC construction ALRED and a specific instance ALPHA-MAC.
In Henri Gilbert and Helena Handschuh, editors, *FSE 2005*, volume 3557 of *LNCS*, pages 1–17. Springer, February 2005.

Goce Jakimoski and Samant Khajuria.
ASC-1: An authenticated encryption stream cipher.
In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *LNCS*, pages 356–372. Springer, August 2011.

Dmitry Khovratovich and Christian Rechberger.
The LOCAL attack: Cryptanalysis of the authenticated encryption scheme ALE.
In *SAC*, Lecture Notes in Computer Science, 2013.
to appear.

Kazuhiro Suzuki, Dongvu Tonien, Kaoru Kurosawa, and Koji Toyota.
Birthday paradox for multi-collisions.
In Min Surp Rhee and Byoungcheon Lee, editors, *ICISC 06*, volume 4296 of *LNCS*, pages 29–40. Springer, November / December 2006.

Bibliography III

Zheng Yuan, Wei Wang, Keting Jia, Guangwu Xu, and Xiaoyun Wang.
New birthday attacks on some MACs based on block ciphers.
In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 209–230.
Springer, August 2009.