Cryptanalysis of SKINNY in the Framework of the SKINNY 2018-2019 Cryptanalysis Competition

Patrick Derbez[1], Virginie Lallemand[2], Aleksei Udovenko[3]

[1]Univ Rennes, CNRS, IRISA, France

[2]Université de Lorraine, CNRS, Inria, France

[3]SnT and CSC, University of Luxembourg, Luxembourg

SAC 2019

Problem

Given a set of 2^{20} **messages**, **practically** recover the 128-bit keys of reduced versions of SKINNY-64-128 and SKINNY-128-128

$$\text{Skinny}-n-k$$

block size

key size

Overview of SKINNY

SKINNY [BJK+16]

The SKINNY family of block ciphers and its low-latency variant MANTIS
Beierle, Jean, Kölbl, Leander, Moradi, Peyrin, Sasaki, Sasdrich, Sim
Crypto 2016

- Performs as well as Simon
- Follows the Tweakey Framework [JNP14] :

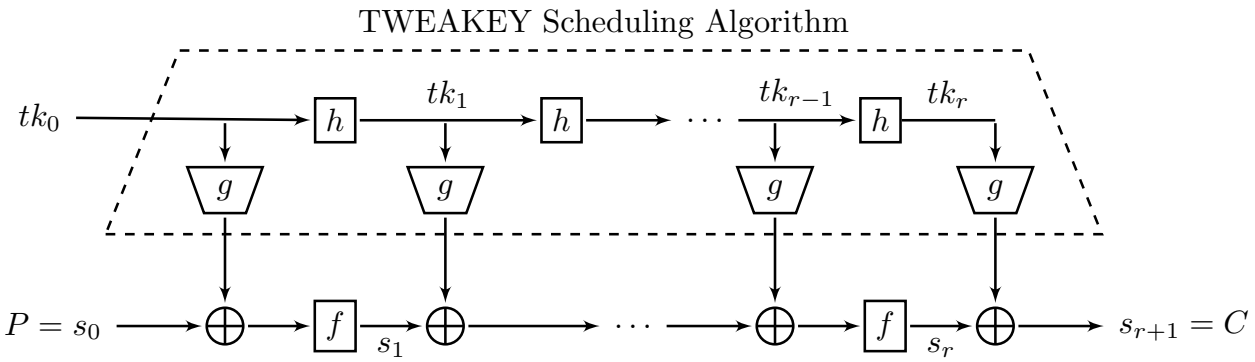
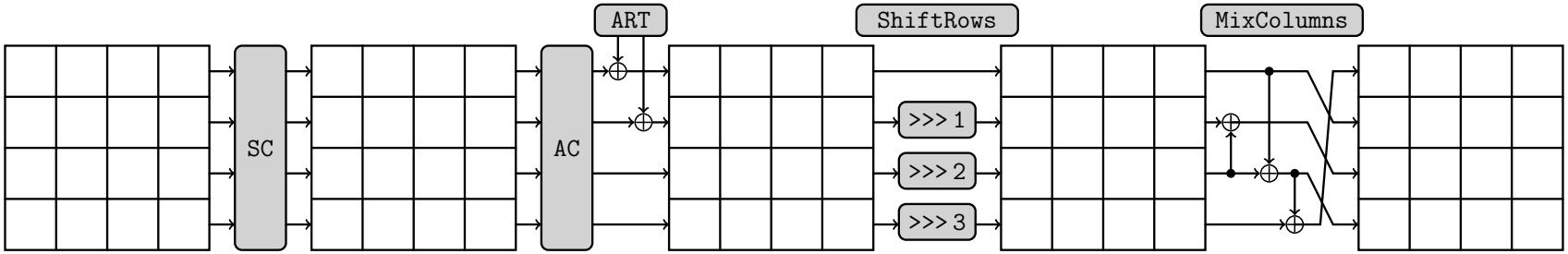


Figure credits: TikZ for Cryptographers [Jea16]

Skinny round function



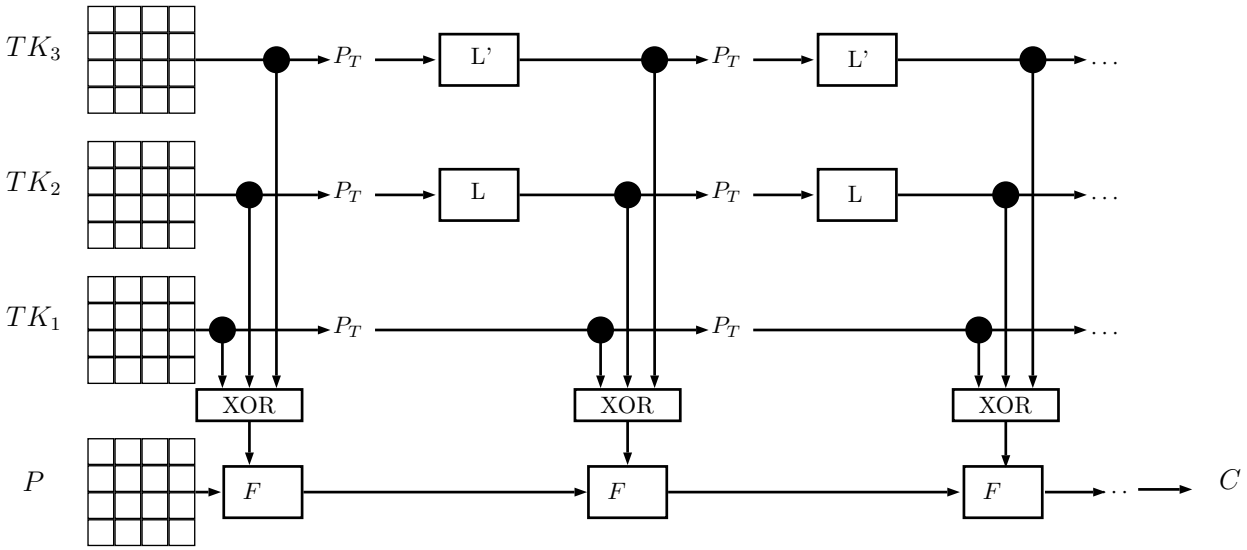
- Block size of 64 or 128 bits
- Tweakey added on the first two lines of the state, **after** SC
- Binary diffusion matrix

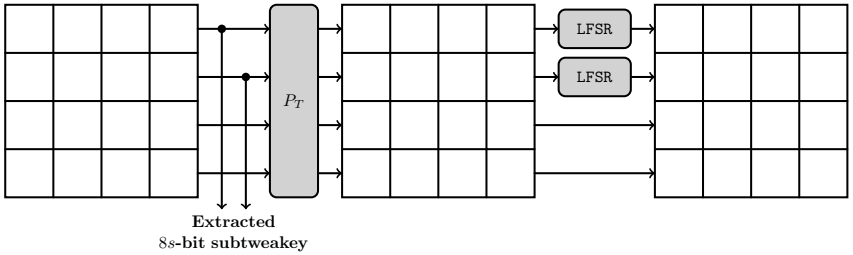
MixColumns Matrix:

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

Figure credits: TikZ for Cryptographers [Jea16]

Skinny Tweakey Schedule





	Tweakey blocks		
block size	1	2	3
$n = 64$	32	36	40
$n = 128$	40	48	56

Figure credits: TikZ for Cryptographers [Jea16]

The SKINNY Competition

The SKINNY 2018-2019 Cryptanalysis Competition

2016 – 2017 Attack small-scaled variants SKINNY-64-128 ($\geq$ 18 rounds), and of SKINNY-128-128 ($\geq$ 22 rounds): [ABC+17] [LGS17] single-key and related-key attacks

2017 – 2018 Similar, except with higher number of rounds

2018 – 2019 More practical scenario:
Provided:

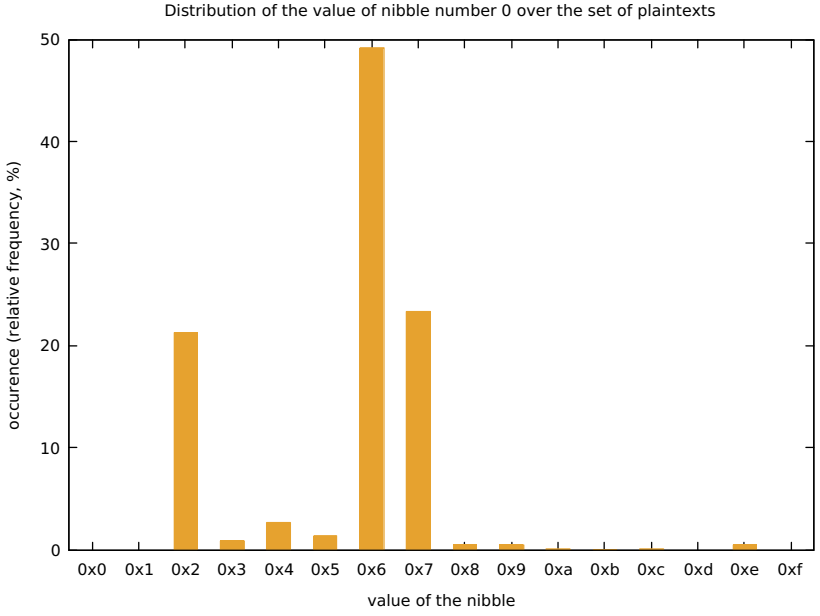
- Set of 2^{20} (plaintexts, ciphertexts) encrypted under a single and secret key
- Sample C code

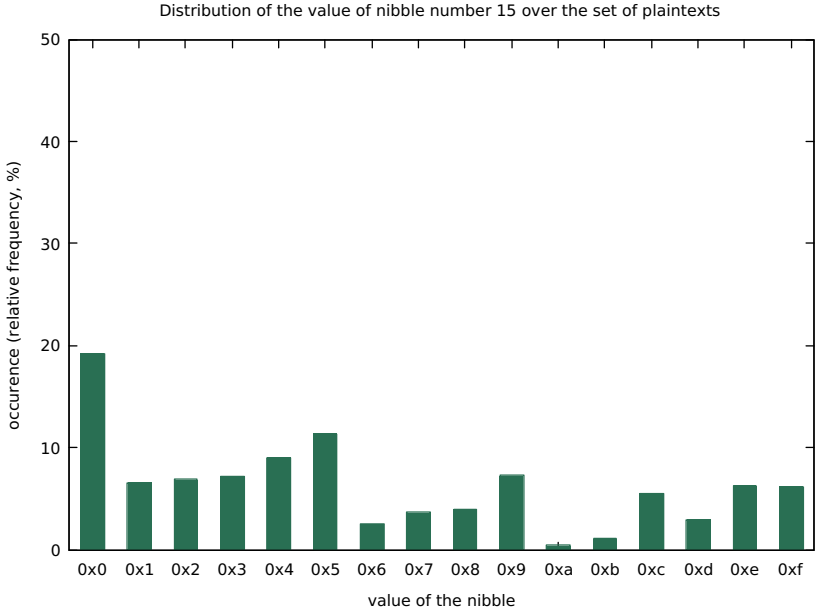
Return the key

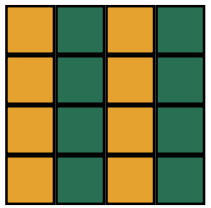
On the Provided Messages

Bias on the Provided Messages

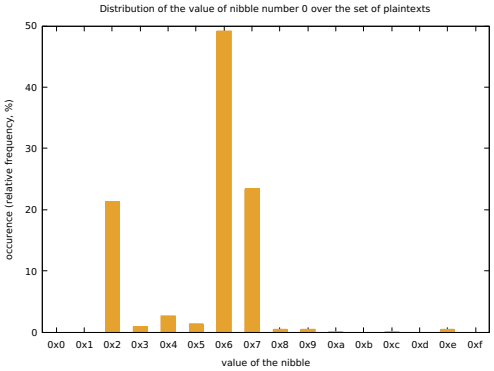
Distribution of the value of nibble 0 (left) and of nibble 15 (right) of the plaintexts for the 12-round attack on SKINNY-64-128

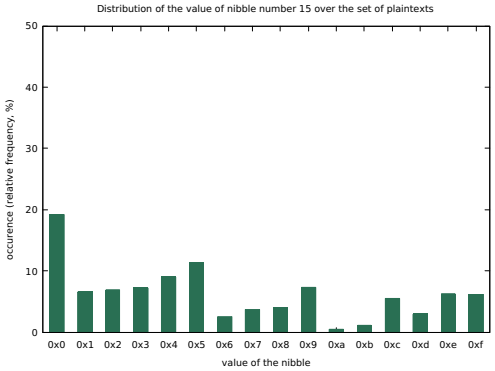




$P =$ 

Recalling the ASCII/UTF8 encoding





	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
⋮								...								
2	space	!	”	#	$	%	&	‘	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	¡	=	¿	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	ˆ	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	—	}	˜	
⋮																

The Plaintexts actually come from English Novels!

> *Project Gutenberg's Alice's Adventures in Wonderland, by Lewis Carroll This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever.*

And few lines later:

> *[...] when suddenly a White Rabbit with pink eyes ran close by her. There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, 'Oh dear! Oh dear! I shall be late!'*

Other data sets correspond to other books (for instance Metamorphosis, by Franz Kafka or The Prince, by Nicolo Machiavelli).

Possible Attacks?

Given our specific set:

- We expect pairs that differ only in few cells
- A **differential** attack seems possible
- Still, we expect that only little data is exploitable: look for **truncated**, high probability distinguisher

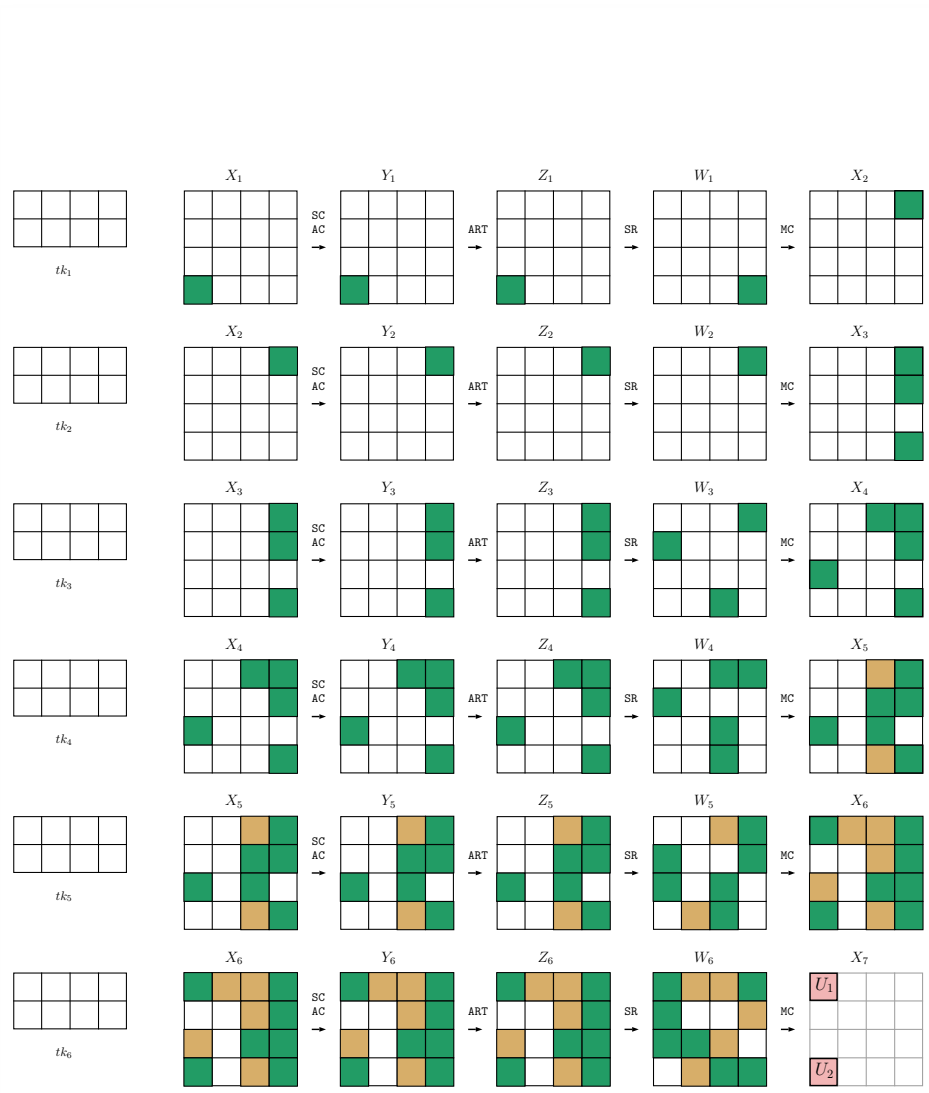
Our Attacks

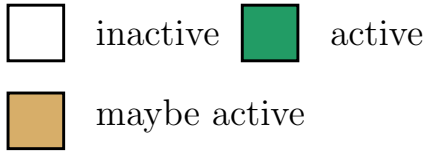
Our Attack on 12-round SKINNY-64-128

- **6 rounds** of truncated differential of **probability 1**
- **1 round** prepended for **free**
- **5 rounds** of key recovery

Total complexity: $2^{51.95}$ basic operations, 32 pairs, 256G memory

Probability 1 Distinguisher over 7 rounds

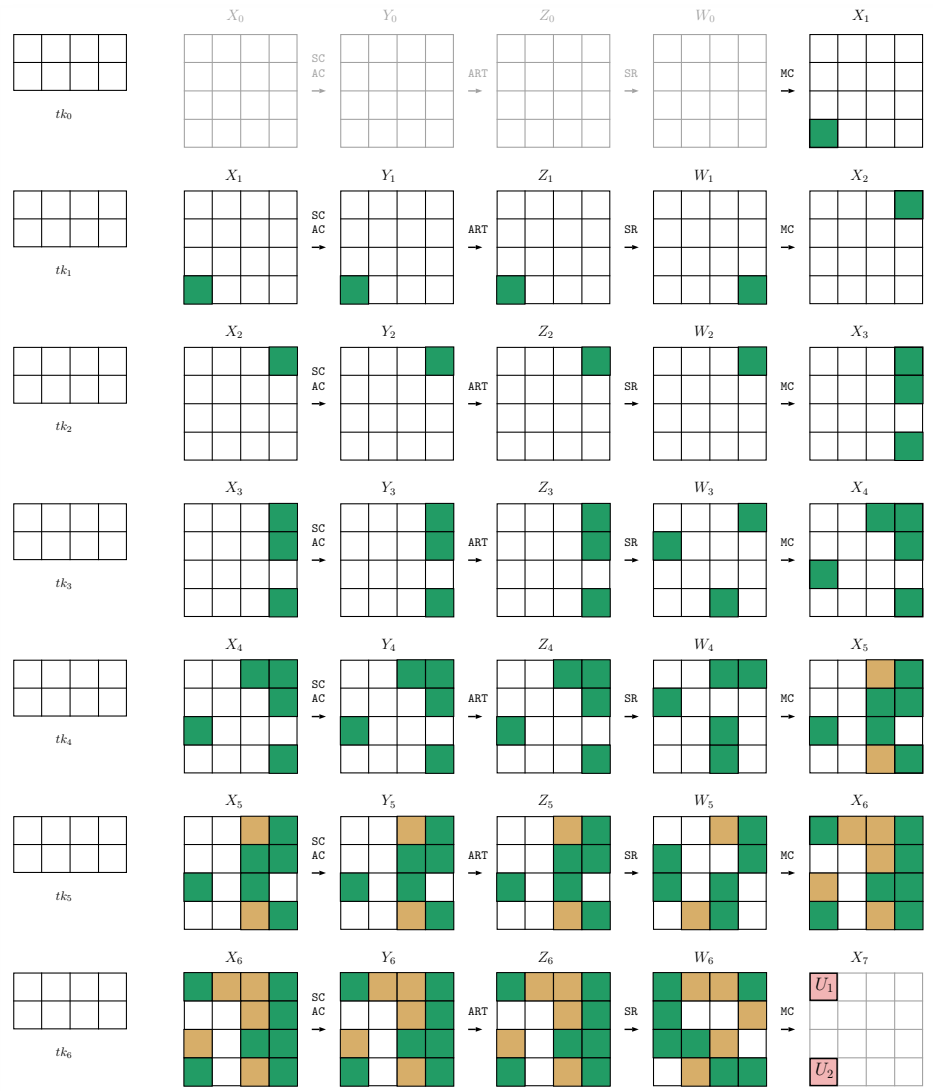




- Truncated differential of probability 1
- If only $X_1[12]$ is active, $X_7[0] = X_7[12]$:

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \end{pmatrix} = \begin{matrix} x_1 + x_3 \\ x_1 \\ x_2 + x_3 \\ x_1 + x_3 \end{matrix}$$

Probability 1 Distinguisher over 7 rounds



inactive active

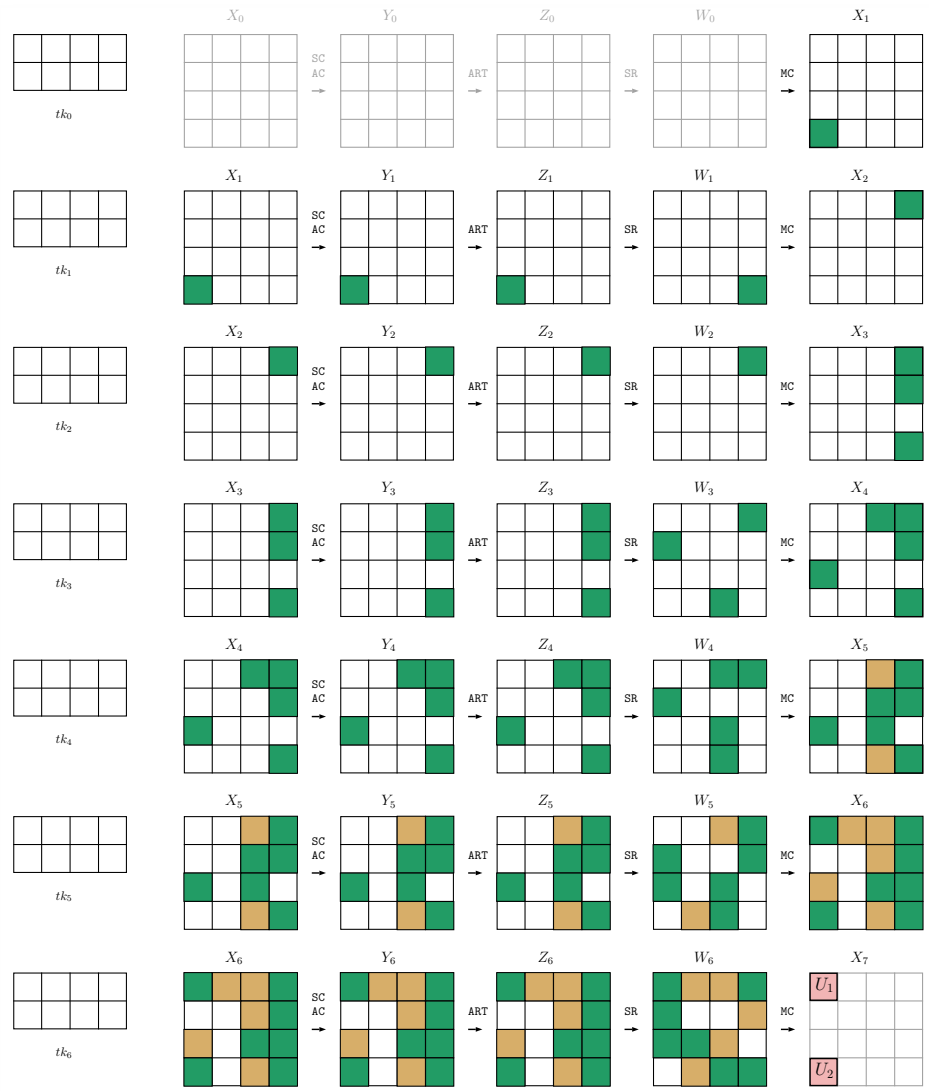
maybe active

- Truncated differential of probability 1
- If only $X_1[12]$ is active, $X_7[0] = X_7[12]$:

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \end{pmatrix} = \begin{matrix} x_1 + x_3 \\ x_1 \\ x_2 + x_3 \\ x_1 + x_3 \end{matrix}$$

- 1 round for free from ARK position

Probability 1 Distinguisher over 7 rounds



inactive active

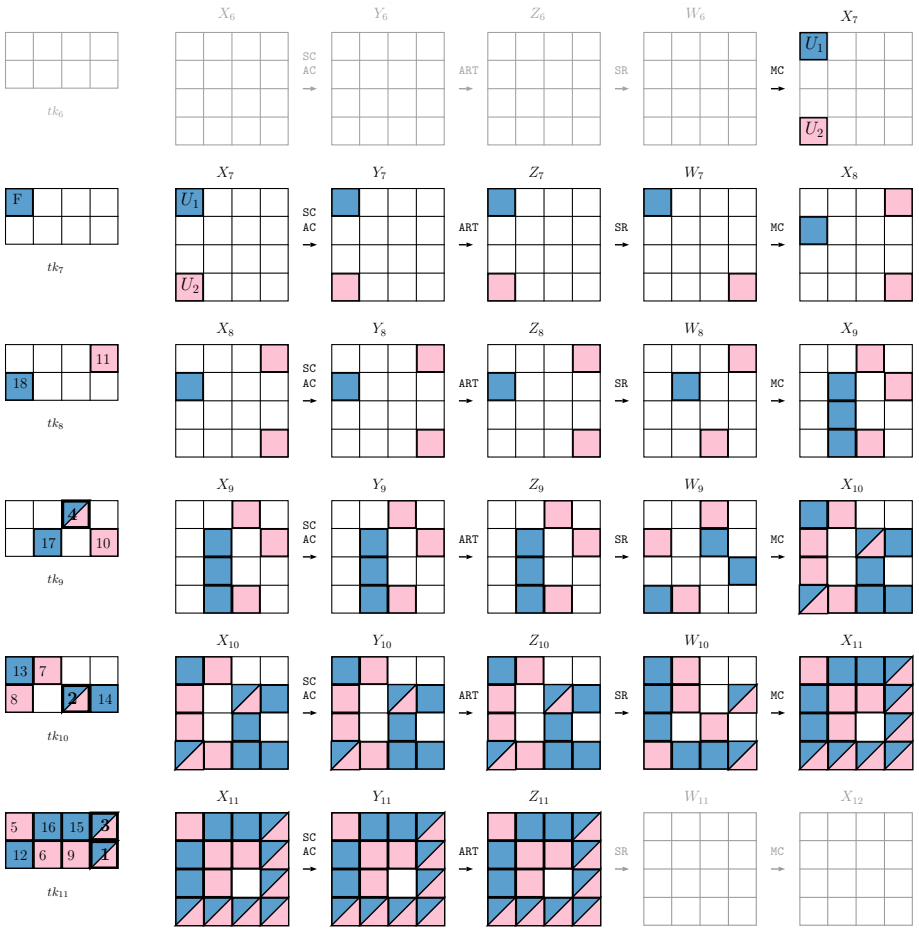
maybe active

- Truncated differential of probability 1
- If only $X_1[12]$ is active, $X_7[0] = X_7[12]$:

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \end{pmatrix} = \begin{matrix} x_1 + x_3 \\ x_1 \\ x_2 + x_3 \\ x_1 + x_3 \end{matrix}$$

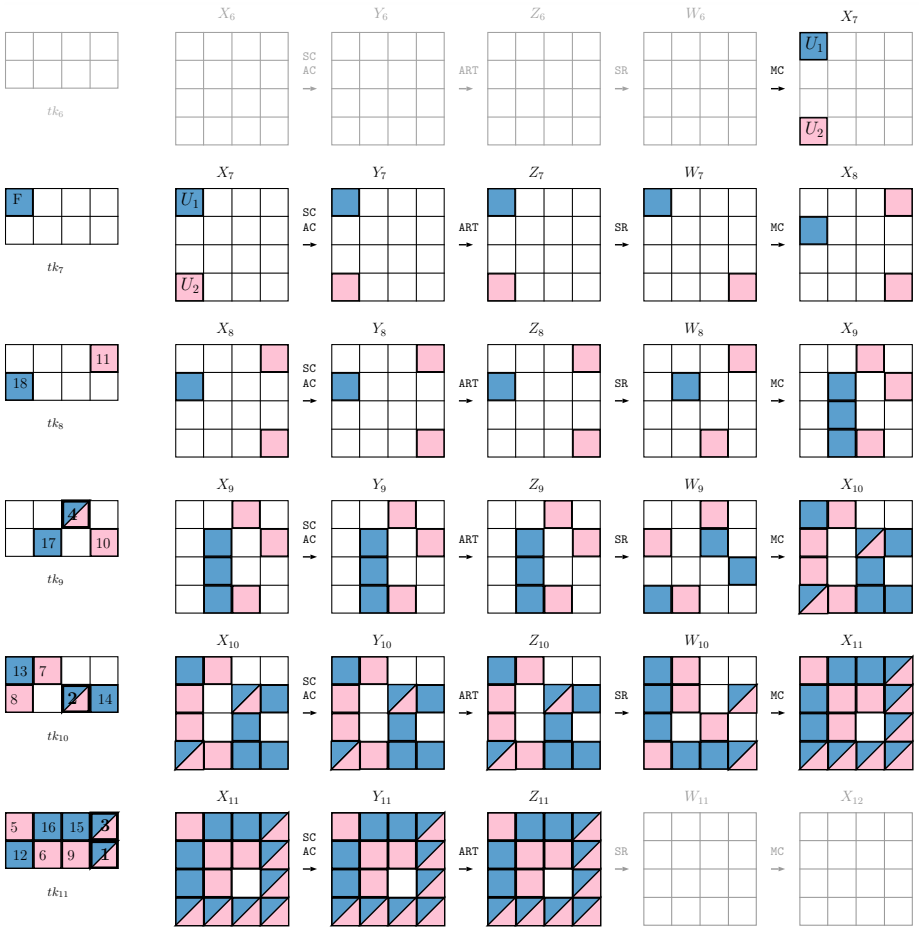
- 1 round for free from ARK position
- In the provided set, 57 pairs follow this trail

Adding 5 rounds of Key Recovery



- in blue the nibbles that are required to compute $X_7[0]$
- in pink the nibbles that are required to compute $X_7[12]$
- total of 19 nibbles, 4 in common

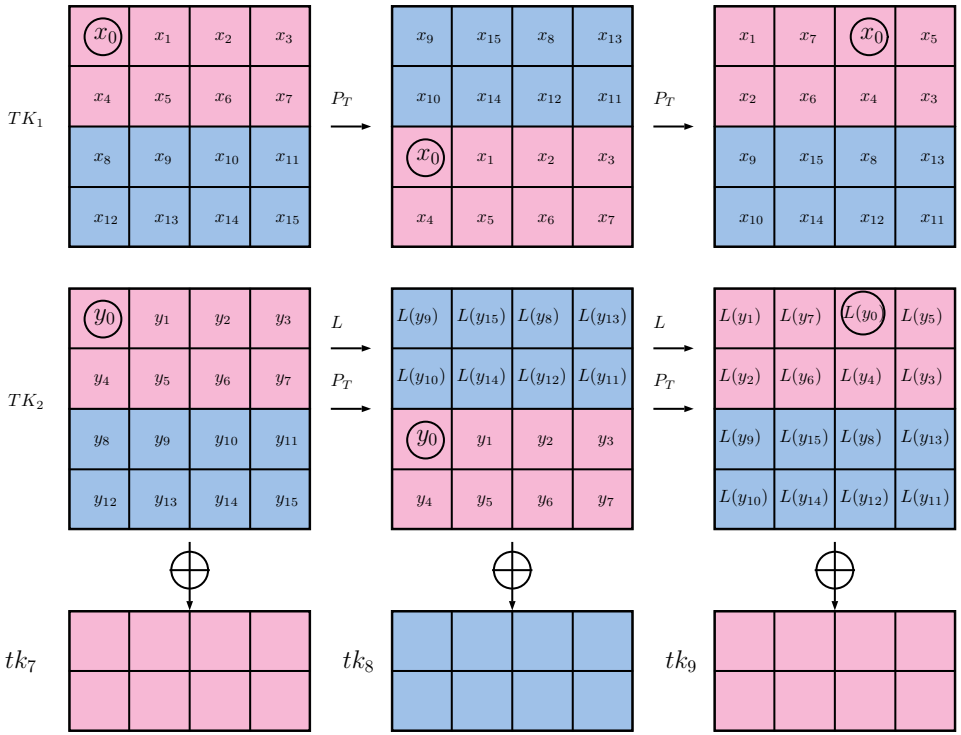
Adding 5 rounds of Key Recovery



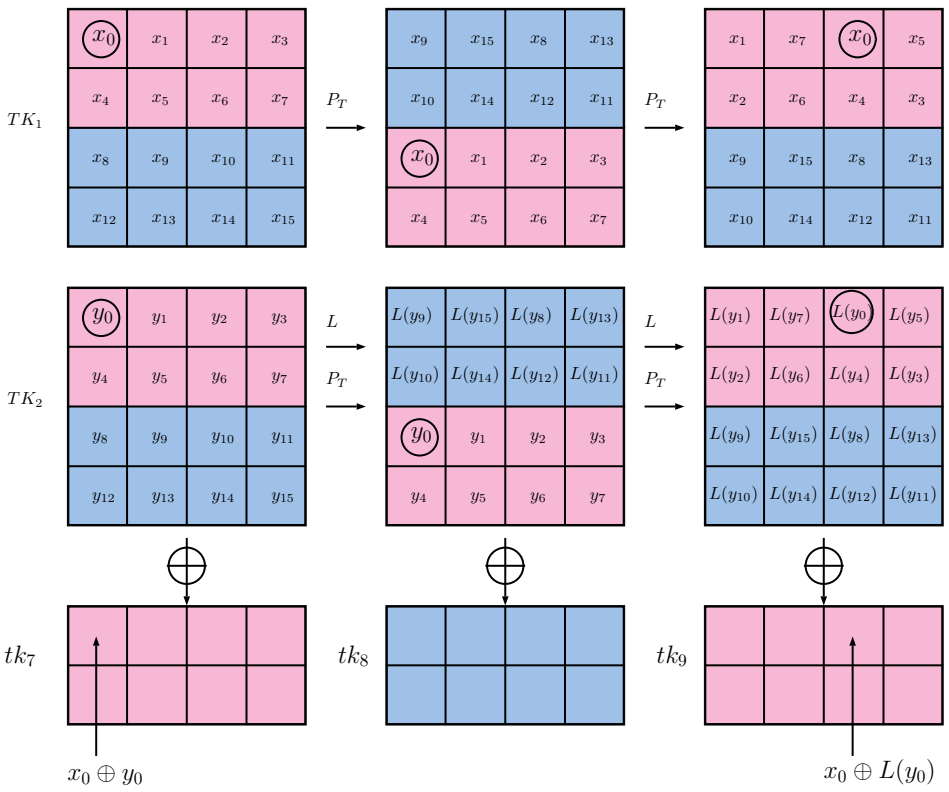
- in blue the nibbles that are required to compute $X_7[0]$
- in pink the nibbles that are required to compute $X_7[12]$
- total of 19 nibbles, 4 in common

1 guess can be saved!

Linear Equations in the TweaKey Schedule (TK2)

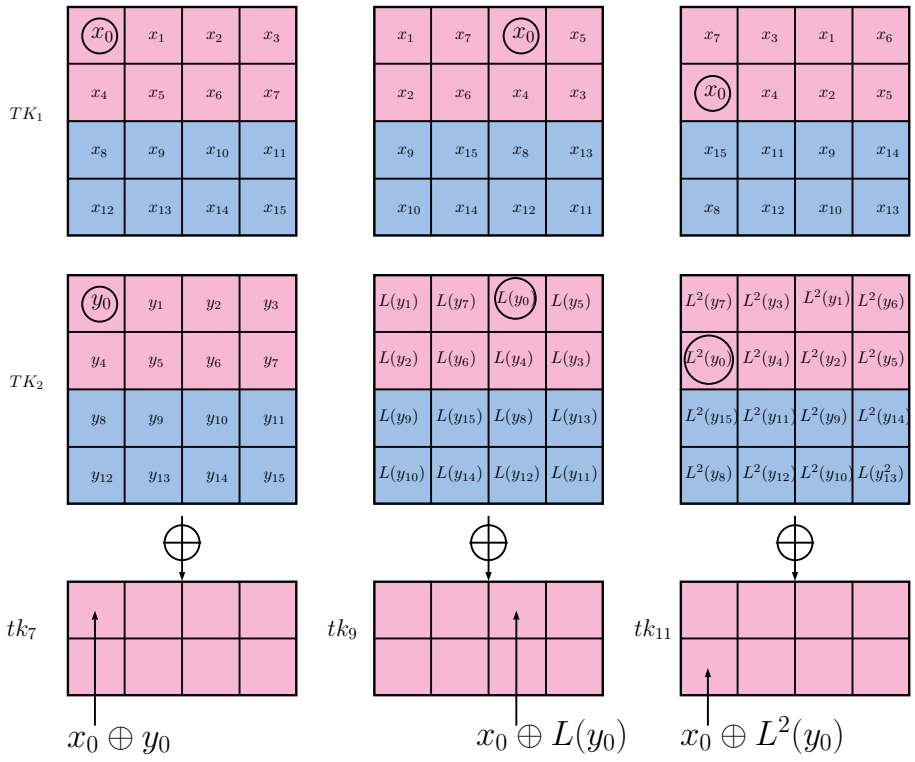


Linear Equations in the TweaKey Schedule (TK2)



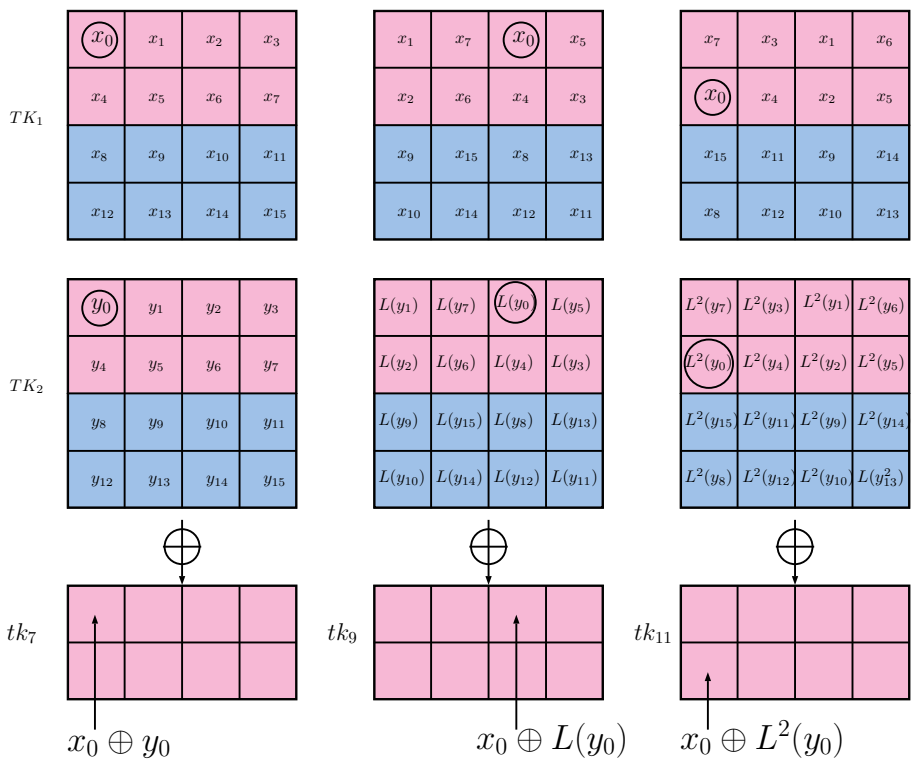
- Tweakey nibbles stay 'aligned' in TK1 and TK2 (e.g. x_0 and y_0)

Linear Equations in the TweaKey Schedule (TK2)



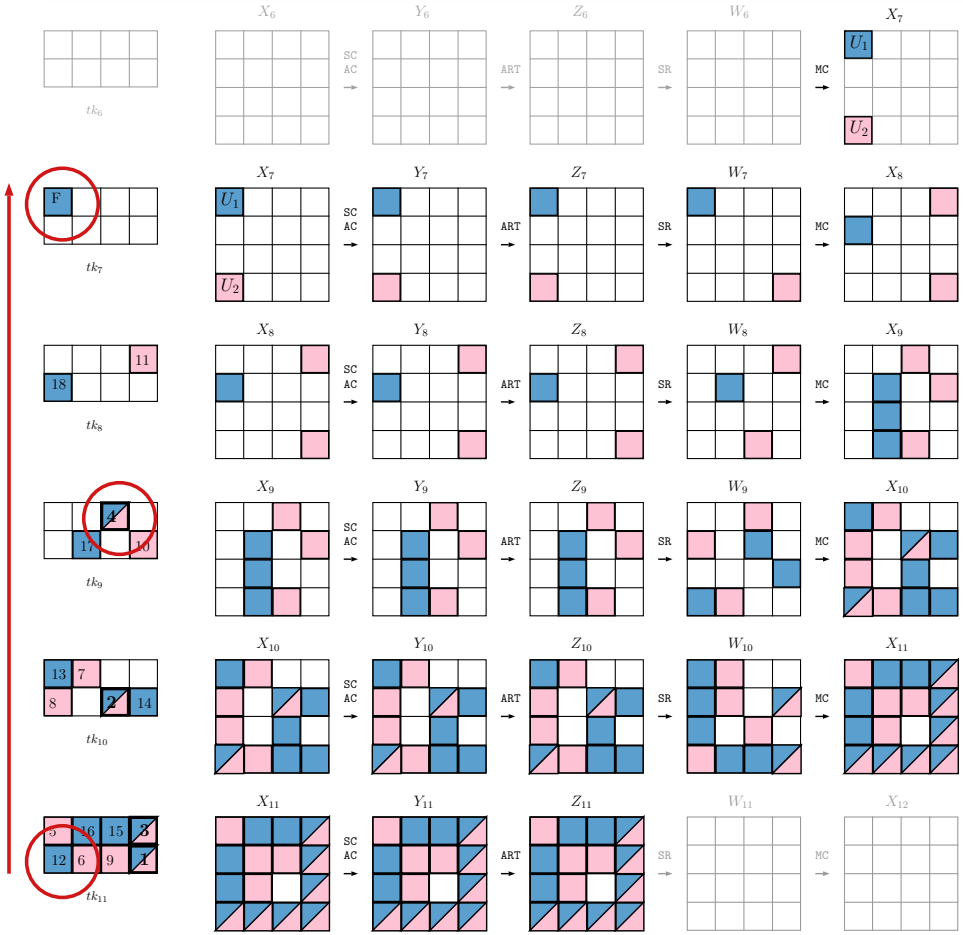
- Focus on odd rounds:
  - $tk_7[0] = x_0 \oplus y_0$
  - $tk_9[2] = x_0 \oplus L(y_0)$
  - $tk_{11}[4] = x_0 \oplus L^2(y_0)$

Linear Equations in the TweaKey Schedule (TK2)



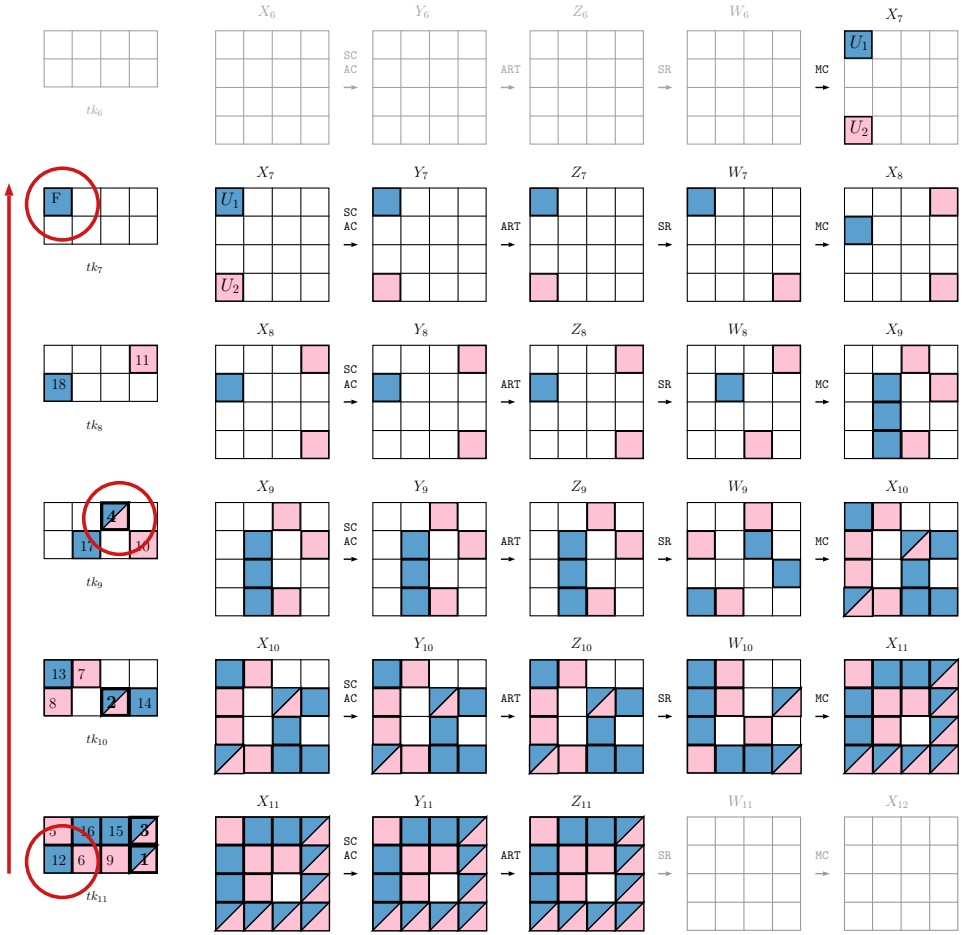
- Focus on odd rounds:
  - $tk_7[0] = x_0 \oplus y_0$
  - $tk_9[2] = x_0 \oplus L(y_0)$
  - $tk_{11}[4] = x_0 \oplus L^2(y_0)$
- $tk_7[0] = L^{-1}(tk_9[2] \oplus tk_{11}[4]) \oplus tk_9[2]$

Linear Equations in the TweaKey Schedule (TK2)



- Focus on odd rounds:
  - $tk_7[0] = x_0 \oplus y_0$
  - $tk_9[2] = x_0 \oplus L(y_0)$
  - $tk_{11}[4] = x_0 \oplus L^2(y_0)$
- $tk_7[0] = L^{-1}(tk_9[2] \oplus tk_{11}[4]) \oplus tk_9[2]$

Linear Equations in the TweaKey Schedule (TK2)



- Focus on odd rounds:
  - $tk_7[0] = x_0 \oplus y_0$
  - $tk_9[2] = x_0 \oplus L(y_0)$
  - $tk_{11}[4] = x_0 \oplus L^2(y_0)$
- $tk_7[0] = L^{-1}(tk_9[2] \oplus tk_{11}[4]) \oplus tk_9[2]$

$\rightarrow$ If $tk_9[2]$ and $tk_{11}[4]$ are known, $tk_7[0]$ can be deduced

Implementation of the Attack

57 pairs available, 32 used

Implementation of the Attack

57 pairs available, 32 used

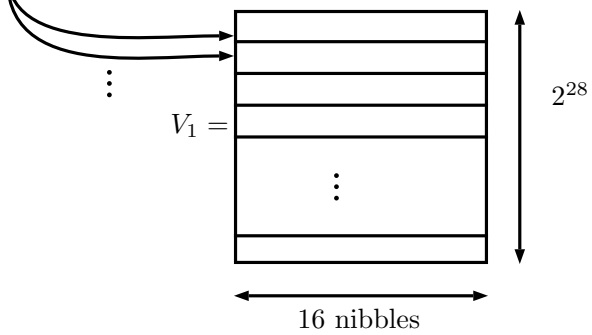
Guess $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$ 

Implementation of the Attack

57 pairs available, 32 used

Guess $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$

Guess $tk_{11}[4], tk_{10}[0], tk_{10}[7], tk_{11}[2]$, $tk_{11}[1], tk_9[5]$ and $tk_8[4]$

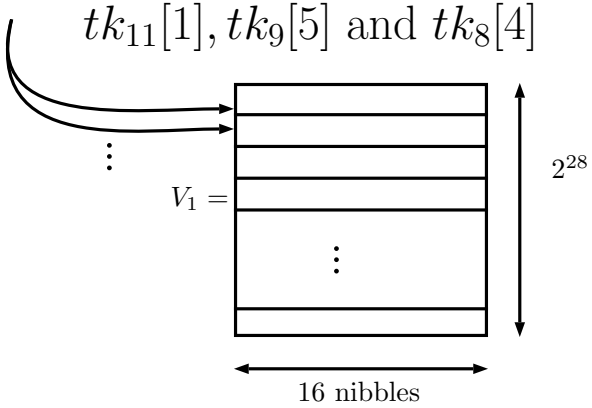


Implementation of the Attack

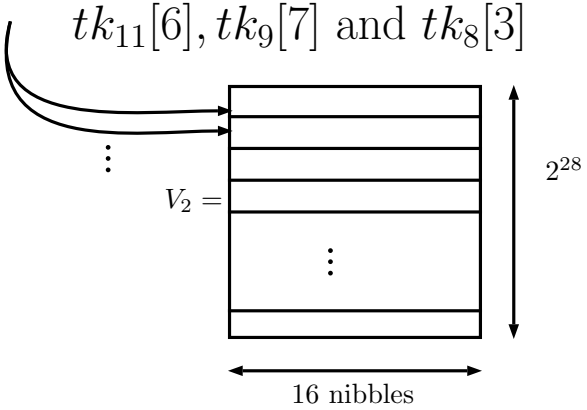
57 pairs available, 32 used

Guess $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$

Guess $tk_{11}[4], tk_{10}[0], tk_{10}[7], tk_{11}[2], tk_{11}[1], tk_9[5]$ and $tk_8[4]$



Guess $tk_{11}[0], tk_{11}[5], tk_{10}[1], tk_{10}[4], tk_{11}[6], tk_9[7]$ and $tk_8[3]$



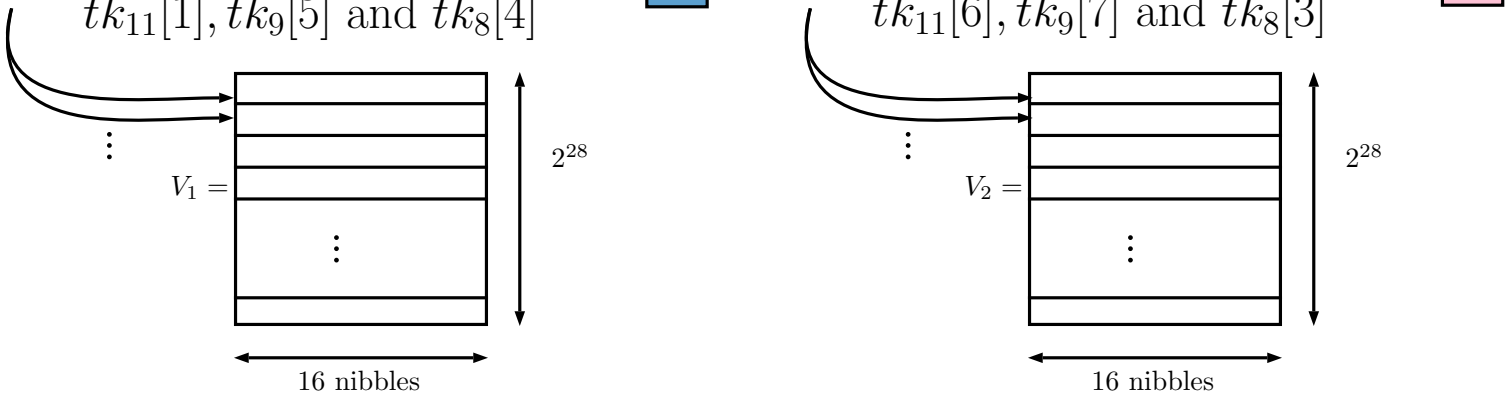
Implementation of the Attack

57 pairs available, 32 used

Guess $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$

Guess $tk_{11}[4], tk_{10}[0], tk_{10}[7], tk_{11}[2], tk_{11}[1], tk_9[5]$ and $tk_8[4]$

Guess $tk_{11}[0], tk_{11}[5], tk_{10}[1], tk_{10}[4], tk_{11}[6], tk_9[7]$ and $tk_8[3]$



Sort and Merge V_1 and V_2 : $2^{28} \times 2^{28} \times 2^{-64} = 2^{-8}$

2^8 values of $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$ survive

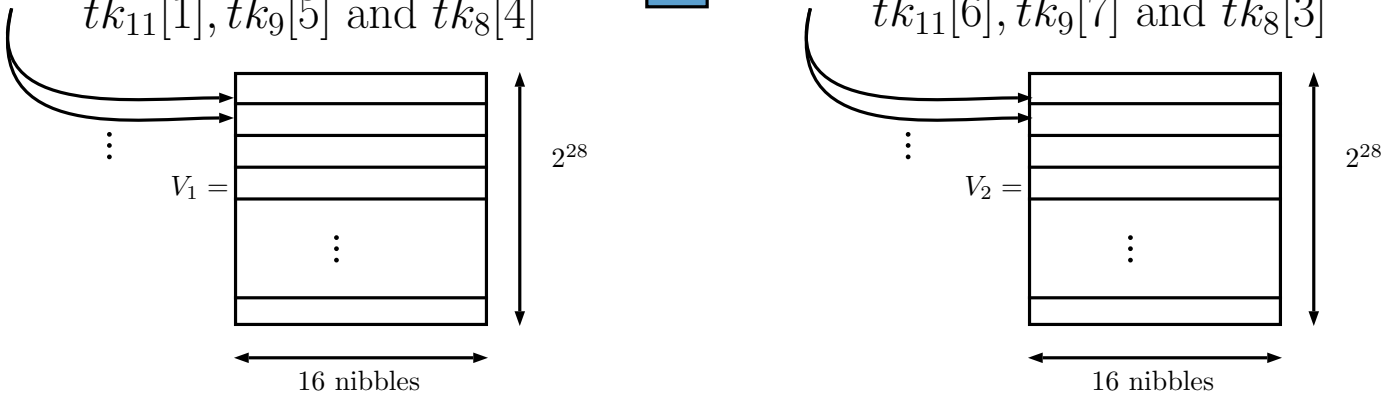
Implementation of the Attack

57 pairs available, 32 used

Guess $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$

Guess $tk_{11}[4], tk_{10}[0], tk_{10}[7], tk_{11}[2], tk_{11}[1], tk_9[5]$ and $tk_8[4]$

Guess $tk_{11}[0], tk_{11}[5], tk_{10}[1], tk_{10}[4], tk_{11}[6], tk_9[7]$ and $tk_8[3]$



Sort and Merge V_1 and V_2 : $2^{28} \times 2^{28} \times 2^{-64} = 2^{-8}$

2^8 values of $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$ survive

for these, repeat with 32 pairs

obtain tk_{11}

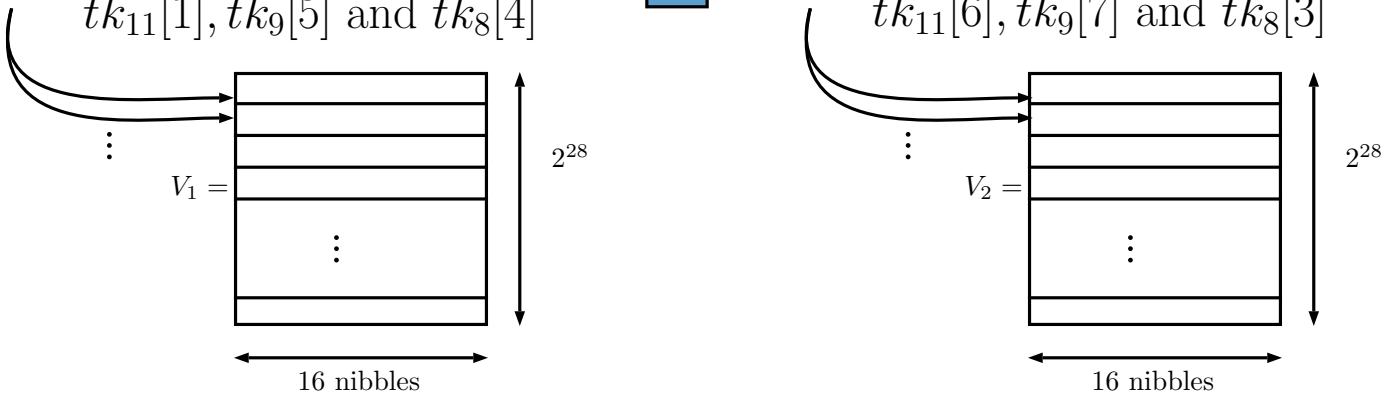
Implementation of the Attack

57 pairs available, 32 used

Guess $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$

Guess $tk_{11}[4], tk_{10}[0], tk_{10}[7], tk_{11}[2], tk_{11}[1], tk_9[5]$ and $tk_8[4]$

Guess $tk_{11}[0], tk_{11}[5], tk_{10}[1], tk_{10}[4], tk_{11}[6], tk_9[7]$ and $tk_8[3]$



Sort and Merge V_1 and V_2 : $2^{28} \times 2^{28} \times 2^{-64} = 2^{-8}$
2^8 values of $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$ survive
for these, repeat with 32 pairs
obtain tk_{11}

$$2^{16} \times (2 \times 32 \times 2^{28} + 2 \times 28 \times 2^{28} + 2^{-8} \times 2 \times 64 \times 2^{28}) \approx 2^{51.95} \text{ op.}$$

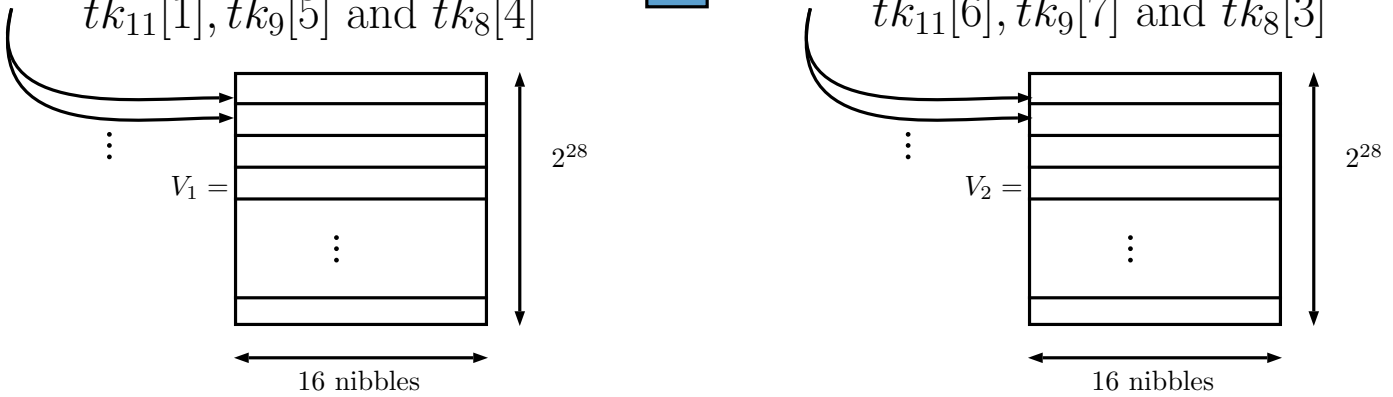
Implementation of the Attack

57 pairs available, 32 used

Guess $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$

Guess $tk_{11}[4], tk_{10}[0], tk_{10}[7], tk_{11}[2], tk_{11}[1], tk_9[5]$ and $tk_8[4]$

Guess $tk_{11}[0], tk_{11}[5], tk_{10}[1], tk_{10}[4], tk_{11}[6], tk_9[7]$ and $tk_8[3]$



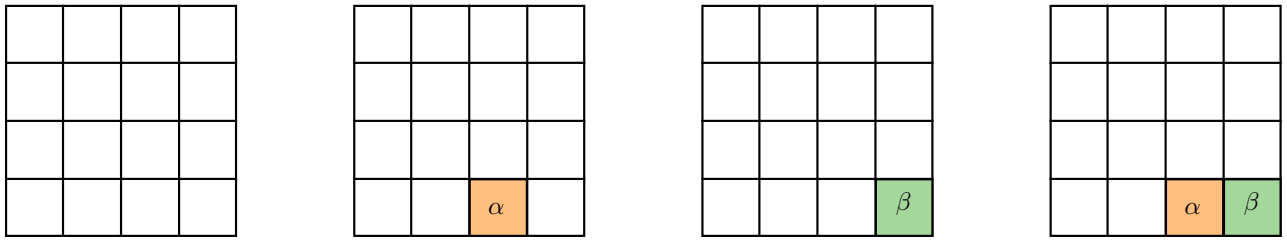
Sort and Merge V_1 and V_2 : $2^{28} \times 2^{28} \times 2^{-64} = 2^{-8}$
2^8 values of $tk_{11}[7], tk_{10}[6], tk_{11}[3]$ and $tk_9[2]$ survive
for these, repeat with 32 pairs
obtain tk_{11}

114 CPU days, 256 GB of memory required

Our Attack on 10-round SKINNY-128-128

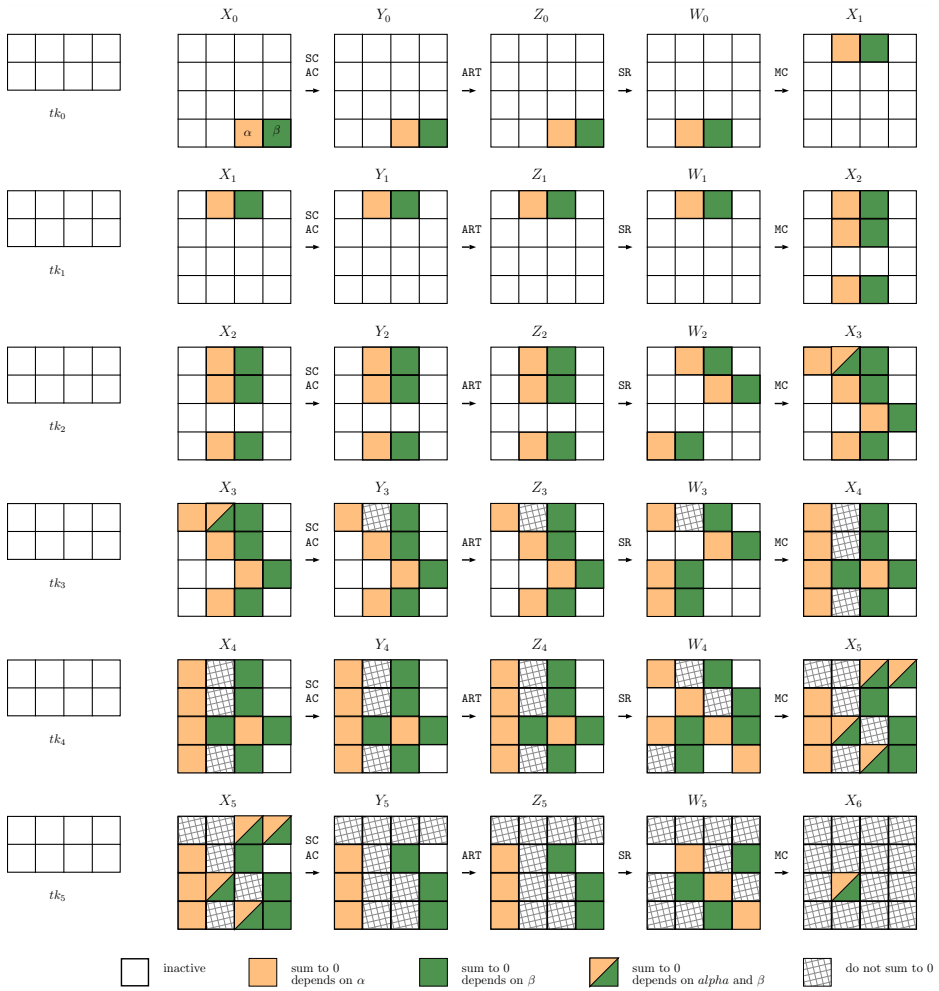
Second order differential [Lai94, Knu95]

- (first-order) differentials consider difference between 2 messages
- (second-order) differentials consider difference between 2^2 messages



After encryption over **6 rounds**, the values obtained for cell 9 sum to 0

Probability 1 Distinguisher over 6 Rounds



- follow the propagation of α (orange) and of β (green)

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \times \begin{matrix} \\ x_2 \\ x_3 \\ \\ \end{matrix}$$

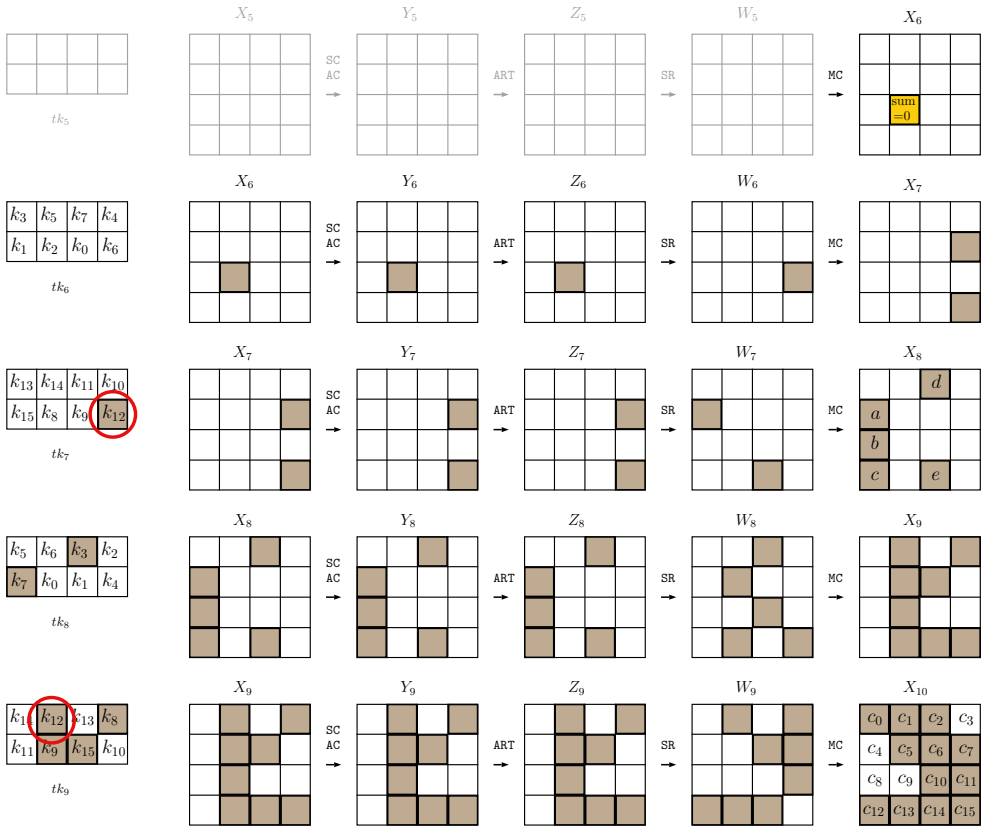
$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \times \begin{matrix} \\ x_2 + f(\alpha) \\ x_3 \\ \\ \end{matrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \times \begin{matrix} \\ x_2 \\ x_3 + g(\beta) \\ \\ \end{matrix}$$

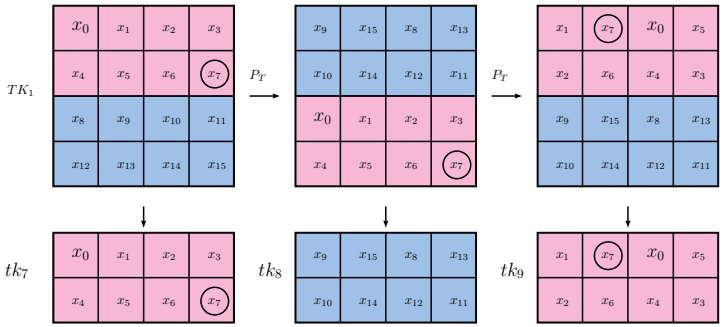
$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \times \begin{matrix} \\ x_2 + f(\alpha) \\ x_3 + g(\beta) \\ \\ \end{matrix}$$

$$\bigoplus \qquad = \begin{matrix} \\ \\ 0 \\ \\ \end{matrix}$$

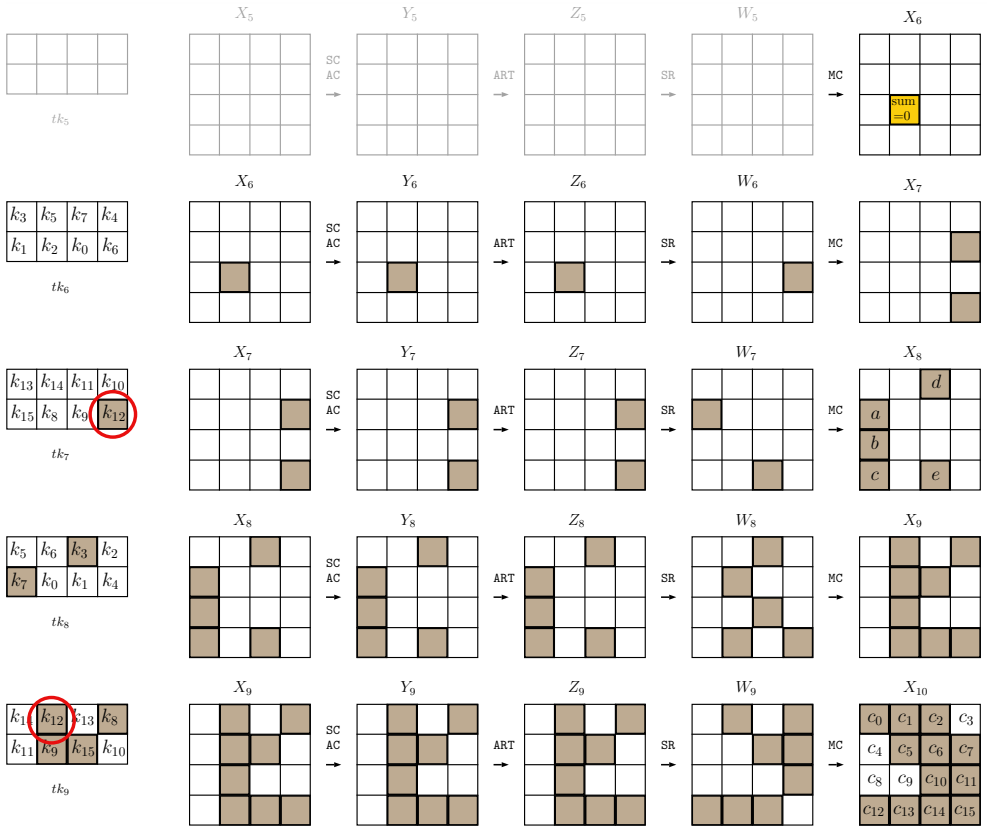
Adding 4 rounds of Key Recovery



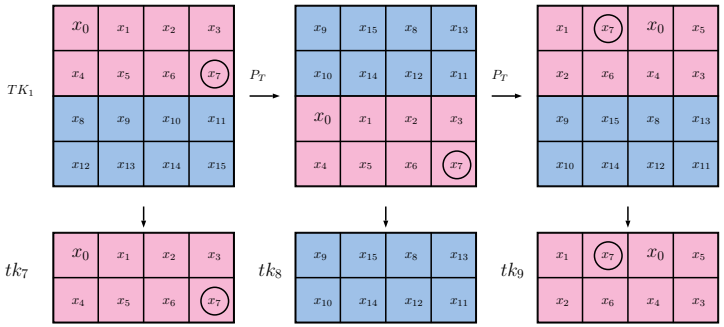
- 7 key bytes involved, 6 unique



Adding 4 rounds of Key Recovery

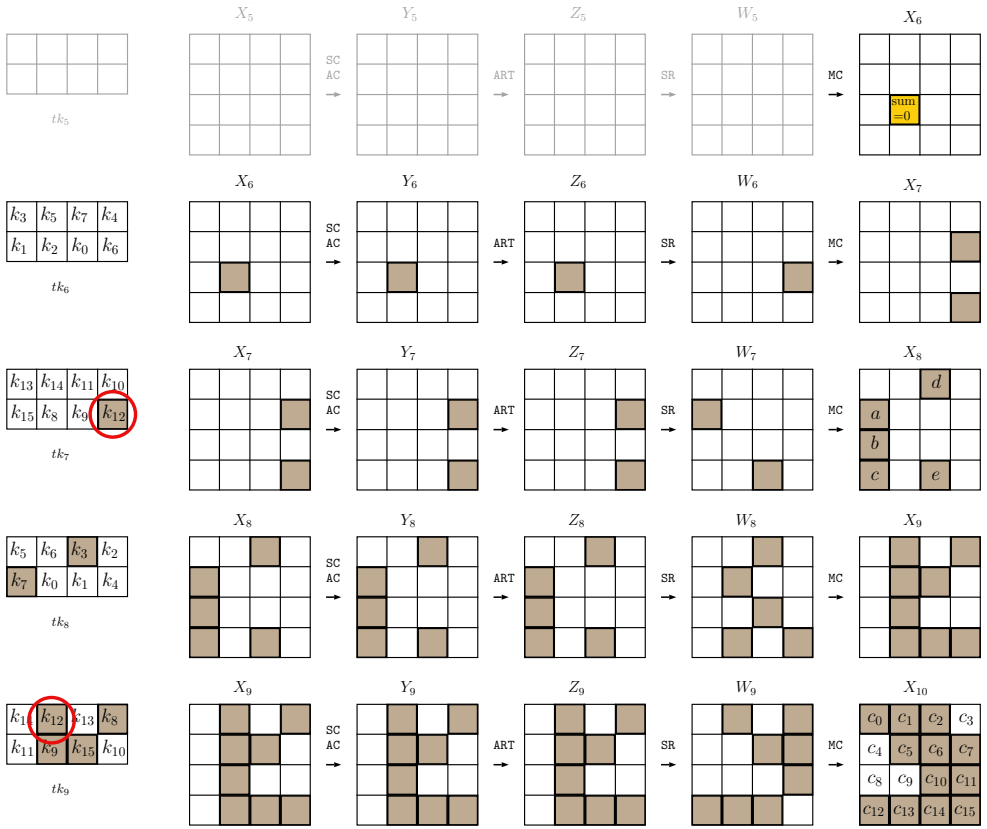


- 7 key bytes involved, 6 unique

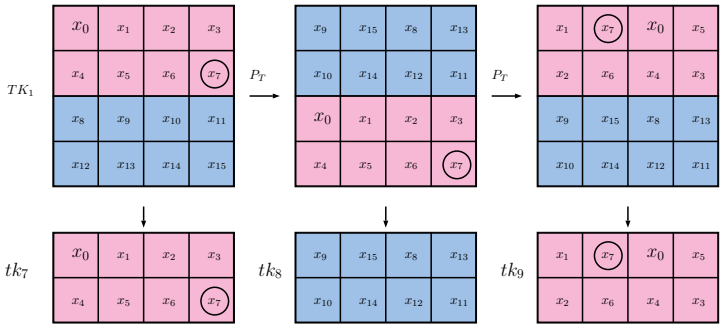


- Guess 6 bytes of key, invert rounds and check that $X_6[9]$ sums to 0
- 6 quadruples are sufficient

Adding 4 rounds of Key Recovery



- 7 key bytes involved, 6 unique



- Guess 6 bytes of key, invert rounds and check that $X_6[9]$ sums to 0
- 6 quadruples are sufficient

2^{52} operations, 32 CPU days, 24 messages, 0.5 GB of memory

Conclusion

- We showed that 12-round SKINNY-64-128 and 10-round SKINNY-128-128 can be attacked in practical time

Version	Rounds	Technique	Data	Time	Memory
SKINNY-64-128	12	Trunc. diff.	64	$2^{51.95}$	256 GB
SKINNY-128-128	10	2nd-order T.diff	24	2^{52}	0.5 GB

- So far these are the best attacks of the challenge
- The challenge is still open and names will still be added to the list for winning any challenge.

Source code available at: http://skinnysac19.gforge.inria.fr/

Conclusion

- We showed that 12-round SKINNY-64-128 and 10-round SKINNY-128-128 can be attacked in practical time

Version	Rounds	Technique	Data	Time	Memory
SKINNY-64-128	12	Trunc. diff.	64	$2^{51.95}$	256 GB
SKINNY-128-128	10	2nd-order T.diff	24	2^{52}	0.5 GB

- So far these are the best attacks of the challenge
- The challenge is still open and names will still be added to the list for winning any challenge.

Source code available at: http://skinnysac19.gforge.inria.fr/

Thank you for your attention

Bibliography I

Ralph Ankele, Subhadeep Banik, Avik Chakraborti, Eik List, Florian Mendel, Siang Meng Sim, and Gaoli Wang.

Related-key impossible-differential attack on reduced-round skinny.

In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *ACNS 17: 15th International Conference on Applied Cryptography and Network Security*, volume 10355 of *Lecture Notes in Computer Science*, pages 208–228. Springer, Heidelberg, July 2017.

Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim.

The SKINNY family of block ciphers and its low-latency variant MANTIS.

In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, Heidelberg, August 2016.

Jérémy Jean.

TikZ for Cryptographers.

https://www.iacr.org/authors/tikz/, 2016.

Bibliography II

Jérémy Jean, Ivica Nikolic, and Thomas Peyrin.
Tweaks and keys for block ciphers: The TWEAKEY framework.
In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, Heidelberg, December 2014.

Lars R. Knudsen.
Truncated and higher order differentials.
In Bart Preneel, editor, *Fast Software Encryption – FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, Heidelberg, December 1995.

Xuejia Lai.
Higher order derivatives and differential cryptanalysis.
In Richard E. Blahut, Daniel J. Costello, Ueli Maurer, and Thomas Mittelholzer, editors, *Communications and Cryptography: Two Sides of One Tapestry*, pages 227–233, Boston, MA, 1994. Springer US.

Bibliography III

Guozhen Liu, Mohona Ghosh, and Ling Song.
Security analysis of SKINNY under related-tweakey settings (long paper).
IACR Transactions on Symmetric Cryptology, 2017(3):37–72, 2017.