SKCAM 2025

Improved Cryptanalysis of ChaCha: Beating PNBs with Bit Puncturing

Yosuke Todo (NTT Social Informatics Laboratories)

This is the joint work with Antonio Florez-Gutierrez

History of ChaCha

- ChaCha (since 2008)
  - ARX design.
  - One of the most deployed stream ciphers.
- PNBs: Probabilistic Neural Bits (AFK+08, FSE)
  - PNBs **experimentally** approximate the key-recovery map with the simple one.
- Our (My?) Motivation
  - A cryptographers' in-depth analysis have failed to overcome PNBs.
    It's not convincing that the experimental approaches (PNBs) are the best.
  - At CRYPTO2020, we tackled this problem by **advanced partitioning**,
    but unfortunately, we failed to beat PNBs completely.
  - ***This is "my" rematch!!***

How to beat PNBs

- Puncturing (FT24, Eurocrypt)
  - A novel method to approximate the key-recovery map.
    - Analyze the key-recovery map from its **Walsh spectrum**.
    - Some non-zero coefficients are changed to zero (**puncturing**).
    - Data complexity is increased to compensate puncturing.
  - Better linear attacks against the S-box-based ciphers.
- How about ARX ciphers?
  - Analyzing Walsh spectrum is challenging.
  - A new tool (**trail enumeration puncturing**)
  - We can successfully replace PNBs with puncturing.

Summary of Results

Round	Data	Time	Note	Ref
6	2^73.7	2^75.5	PNBs w/ syncopation	Wang et al., CRYPTO, 2023
	2^41.6	2^71.0	PNBs w/ linear decomposition	Dey, IEEE-IT, 2024
	2^51.0	**2^61.4**		Ours
	2^55.7	**2^57.4**		Ours
7	2^102.6	2^189.7	DL hull and PNBs	Xu et al., ToSC, 2024
	2^127.7	**2^148.2**		Ours
	2^102.9	**2^154.2**		Ours
7.5	2^32.6	2^255.2	PNBs w/ linear decomposition	Dey, IEEE-IT, 2024
	2^127.1	**2^250.2**		Ours

Review of the Existing
What is difficult? What is problem?

ChaCha

$$V^0 = \begin{pmatrix} v_0^0 & v_1^0 & v_2^0 & v_3^0 \\ v_4^0 & v_5^0 & v_6^0 & v_7^0 \\ v_8^0 & v_9^0 & v_{10}^0 & v_{11}^0 \\ v_{12}^0 & v_{13}^0 & v_{14}^0 & v_{15}^0 \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & t_1 & t_2 & t_3 \end{pmatrix}.$$

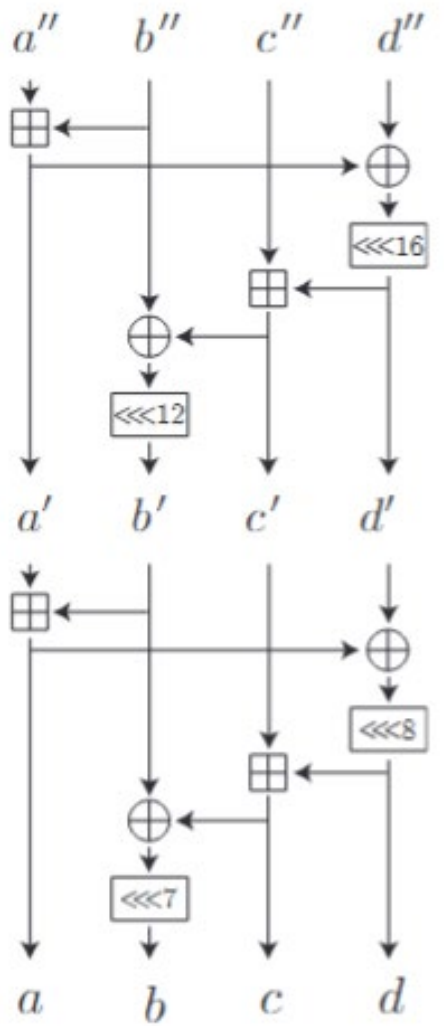
Odd rounds

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix}.$$

Even rounds

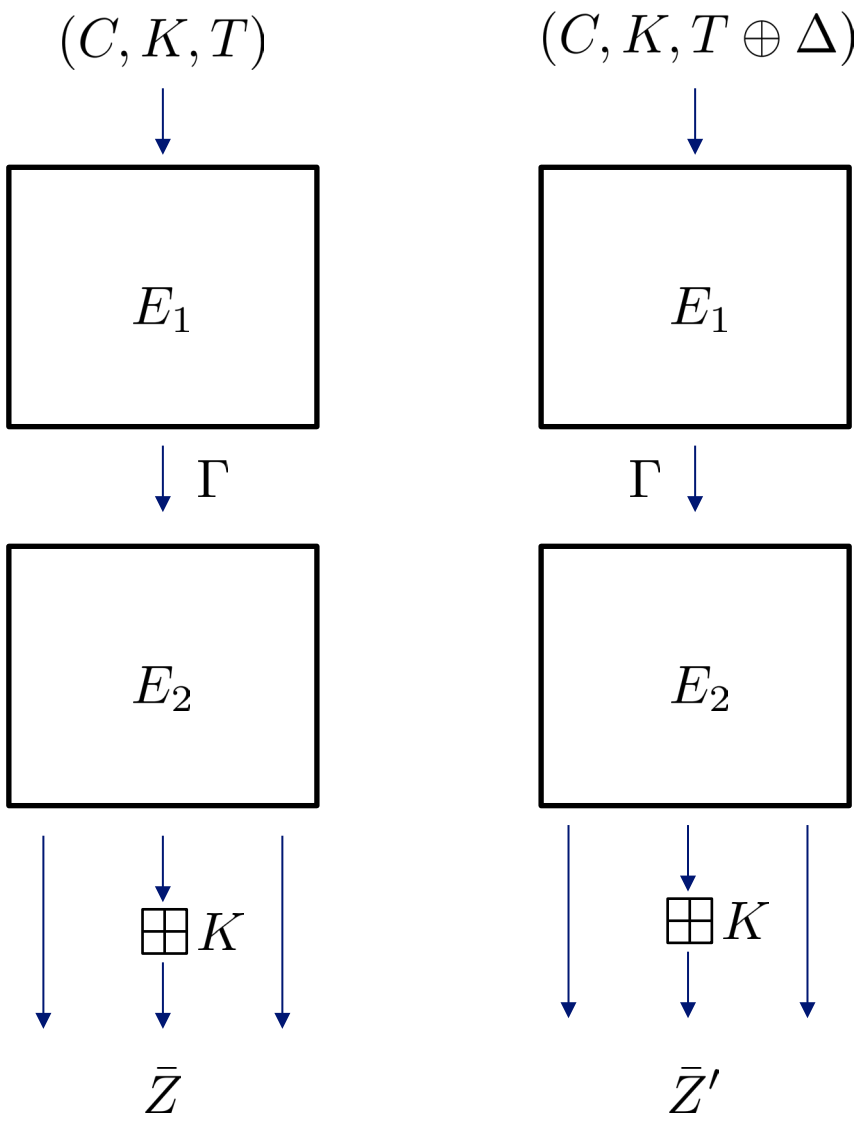
$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix}.$$

$$KS = V^0 + V^R = \begin{pmatrix} v_0^0 + v_0^R & v_1^0 + v_1^R & v_2^0 + v_2^R & v_3^0 + v_3^R \\ v_4^0 + v_4^R & v_5^0 + v_5^R & v_6^0 + v_6^R & v_7^0 + v_7^R \\ v_8^0 + v_8^R & v_9^0 + v_9^R & v_{10}^0 + v_{10}^R & v_{11}^0 + v_{11}^R \\ v_{12}^0 + v_{12}^R & v_{13}^0 + v_{13}^R & v_{14}^0 + v_{14}^R & v_{15}^0 + v_{15}^R \end{pmatrix}$$



$QR(a'', b'', c'', d'')$

Differential-linear attack



Differential-linear distinguisher (aka autocorrelation)

$$\mathrm{Aut}_{E_1}(\Delta, \Gamma) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \Gamma, E_1(x) \rangle \oplus \langle \Gamma, E_1(x \oplus \Delta) \rangle}.$$

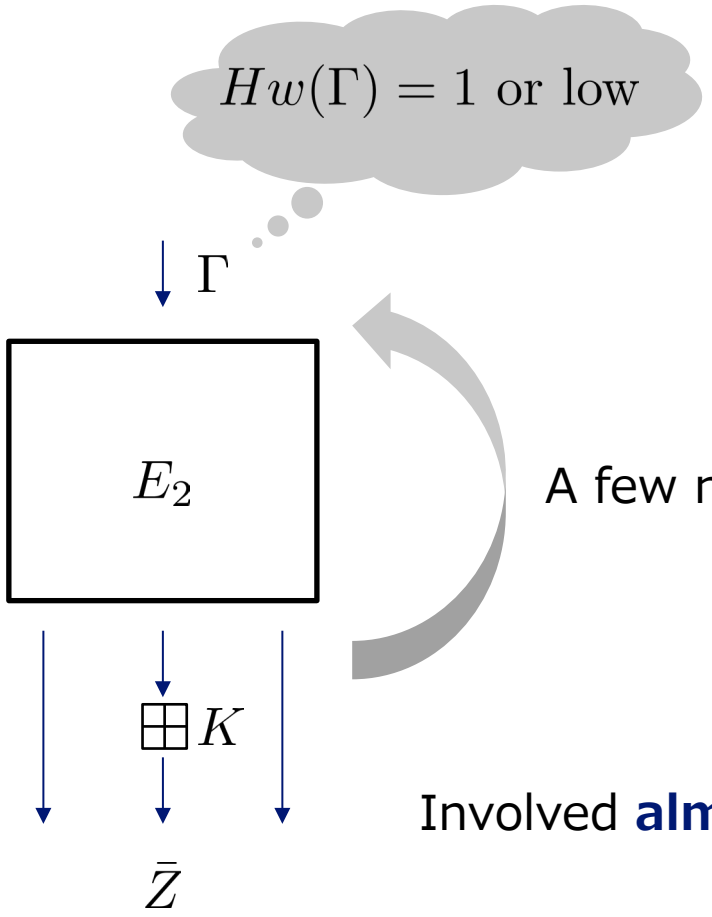
Guess K and check

$$\frac{1}{|\mathbb{X}|} \sum_{x \in \mathbb{X}} (-1)^{\langle \Gamma, E_2^{-1}(\bar{Z} - K) \rangle \oplus \langle \Gamma, E_2^{-1}(\bar{Z}' - K) \rangle}.$$

Correct guess → high correlation
Wrong guess → random (hypothesis)

Key recovery involves many bits quickly

Quick diffusion by ARX



A few rounds

Guess K and check

$$\frac{1}{|\mathbb{X}|}\sum_{x\in\mathbb{X}}(-1)^{\langle\Gamma,E_2^{-1}(\bar{Z}-K)\rangle\oplus\langle\Gamma,E_2^{-1}(\bar{Z}'-K)\rangle}.$$

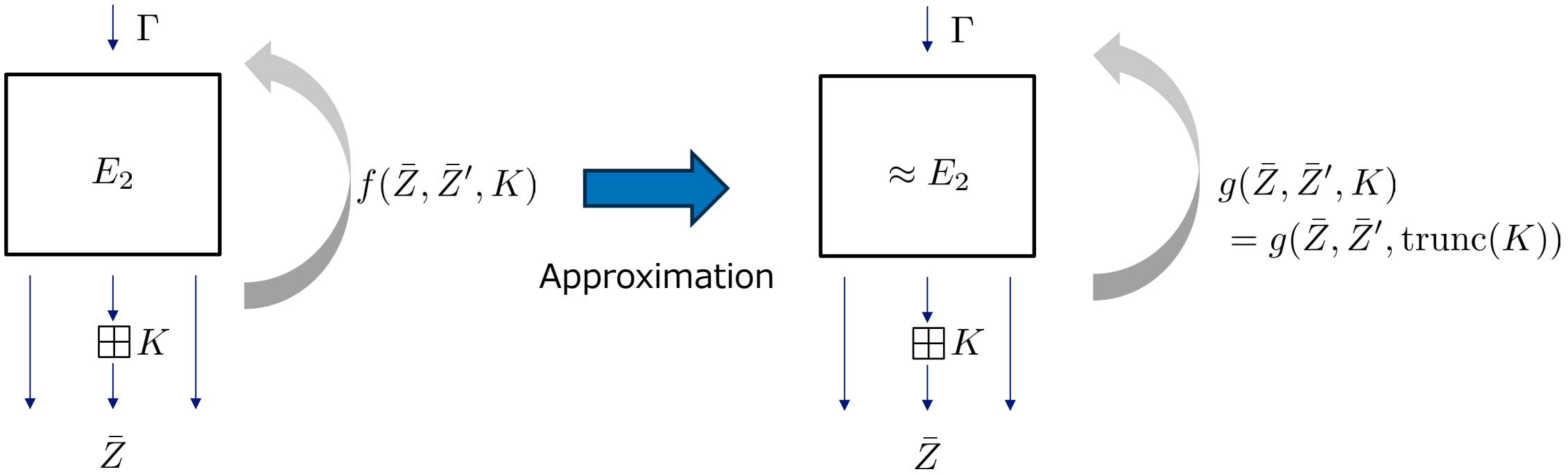
This cost is extremely high.

Involved **almost full bits** of *K*.

PNBs [AFK+08]

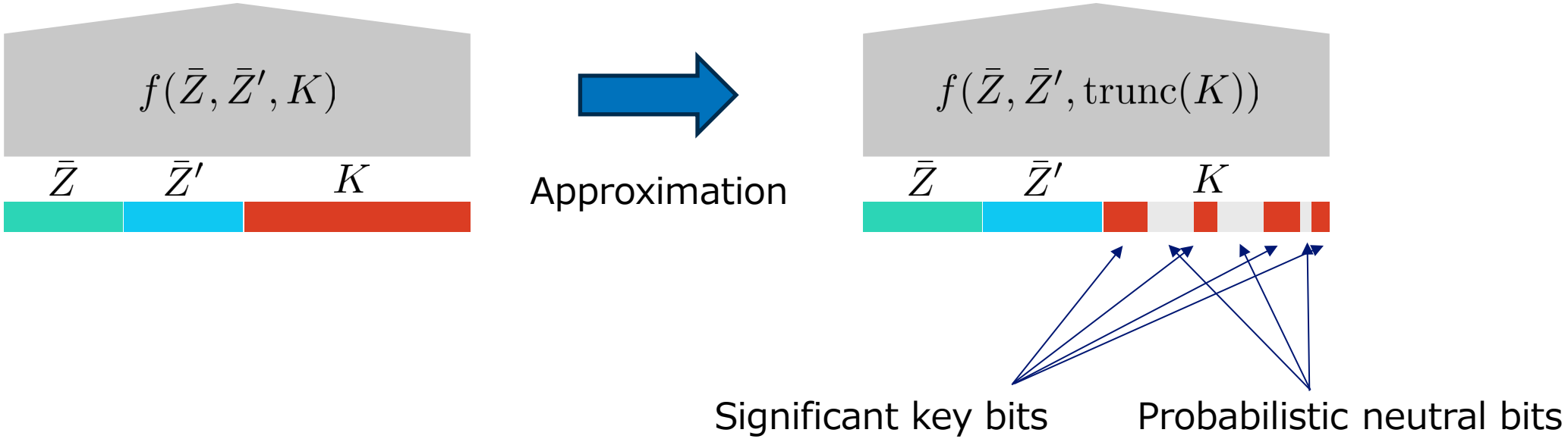
Approximate the key-recovery map.

- $f(\bar{Z}, \bar{Z}', K) \approx g(\bar{Z}, \bar{Z}', K)$ and g doesn't involve many bits of K.
- How to obtain such g?



PNBs [AFK+08]

- K is divided into two parts, guessed bits and PNBs. PNBs are fixed to the constant (usually, all zero)
  - Set of PNBs is experimentally obtained.
  - The final correlation is also experimentally obtained.

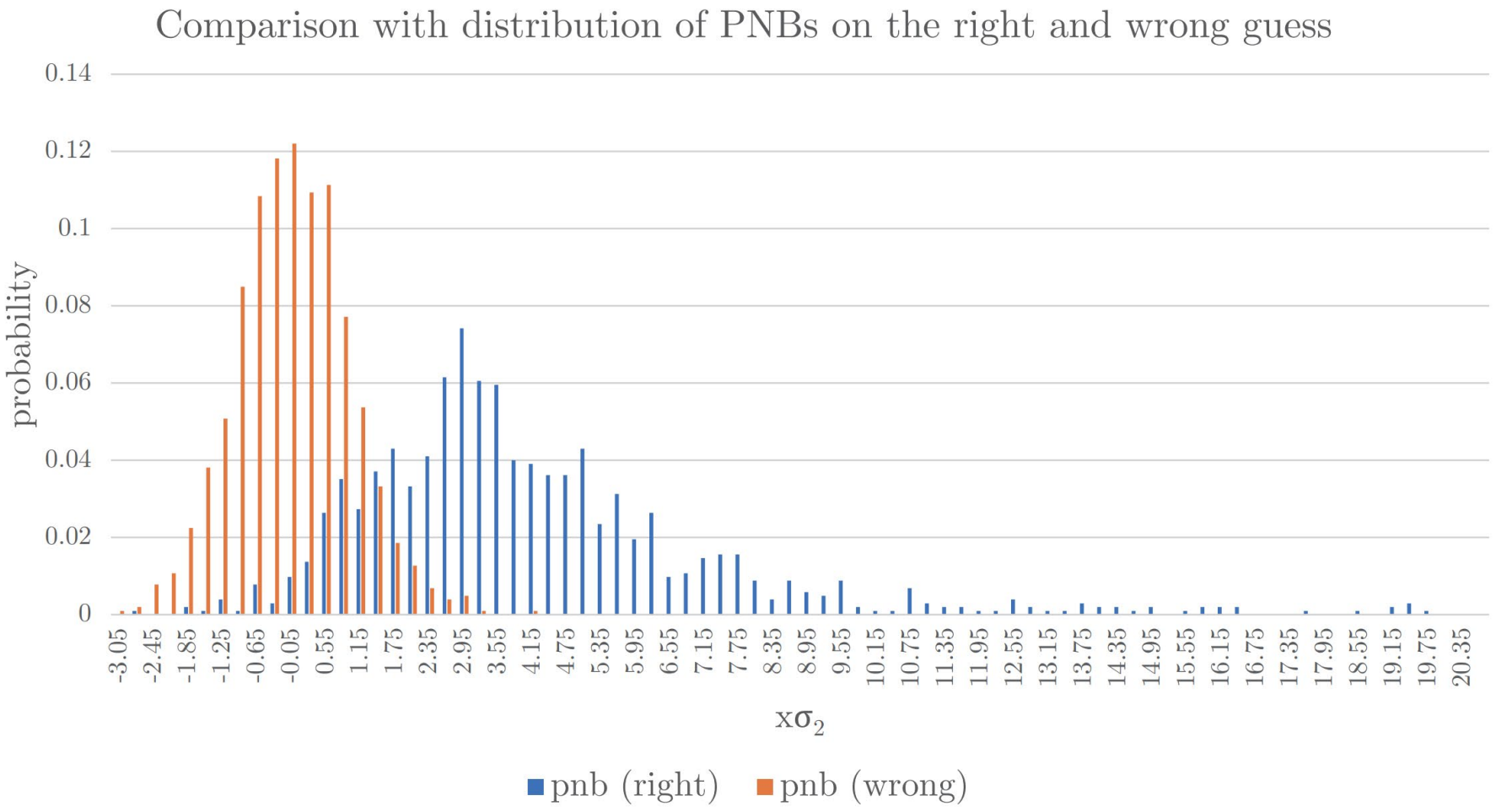


What is problem in PNBs?

PNBs are regarded as blackbox analysis.

- We never analyze the inside of E_2 carefully.
- There is no plausible evidence that we set 0 for PNBs.
  - Some papers suggested 10* is more adequate, but heuristic.
  - Remember PNBs are **"key"**, which is **unknown** for attackers.
    - The attack precision heavily depends on the key.
    - The statistics are never normal.
    - There are significant gap between the average and median.

What is problem in PNBs?



In [AFK+08], the authors recommend to use the median.
Then, the success probability is 50%.

New Theory and New Tool

Mathematical background

Key recovery function. $f : \mathbb{F}_2^n \to \mathbb{F}_2$.

Its Walsh spectrum. $$\widehat{f}(u) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle u,x \rangle} f(x).$$

Correlation. $$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)g(x) = 2^n \langle \hat{f}, \hat{g} \rangle$$

Approximation and penalty (FT24)

Let $f: \mathbb{F}_2^n \to \mathbb{F}_2$ be the original key-recovery function.

We approximate f into pseudoboolean function $g: \mathbb{F}_2^n \to \mathbb{R}$.

Corollary 3(FT24)

If we use g instead of f, to take the same advantage, the sample size is increased by a factor $1/\rho^2$, where

$$\rho = \frac{|\langle f, g \rangle|}{||f||_2 \cdot ||g||_2}$$

Puncturing (FT24)

Walsh spectrum puncturing (FT24).

Let f be a balanced Boolean function and $\hat{f}$ its Walsh spectrum. A puncture set is any subset $\mathcal{P} \subseteq \mathbb{F}_2^n \setminus \{0\}$. We define the punctured function g as

$$\widehat{g}(u) = \begin{cases} \widehat{f}(u) & \text{if } u \notin \mathcal{P}, \\ 0 & \text{if } u \in \mathcal{P}. \end{cases}$$

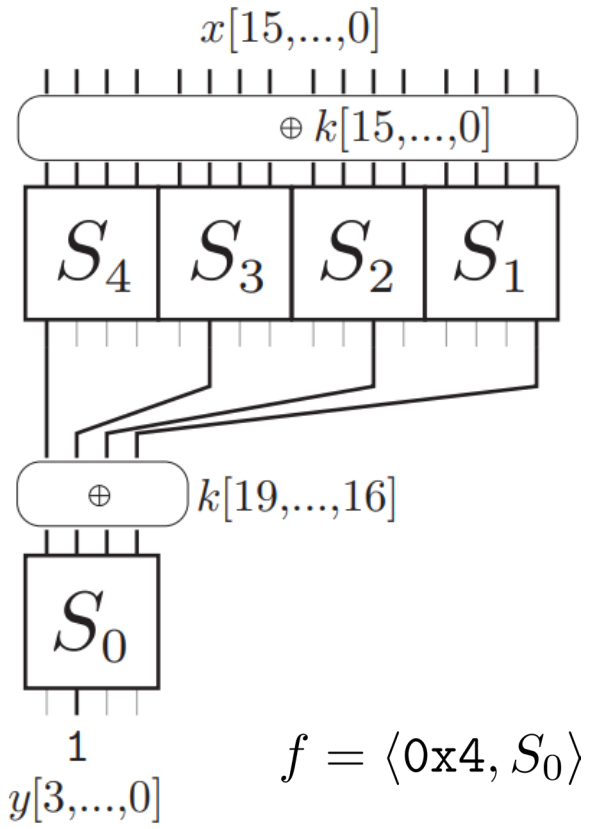
Then, $\rho^2 = \sum_{u \notin \mathcal{P}} \hat{f}(u)^2 = \langle f, g \rangle = cor(f, g)$.

The main idea is analyzing the Walsh spectrum instead of the Boolean function.

Understanding puncturing

Example, PRESENT-like cipher.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
f	1	-1	1	1	-1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1
$\hat{f}$	0	4	0	4	4	0	-4	8	4	0	-4	-8	0	4	0	4



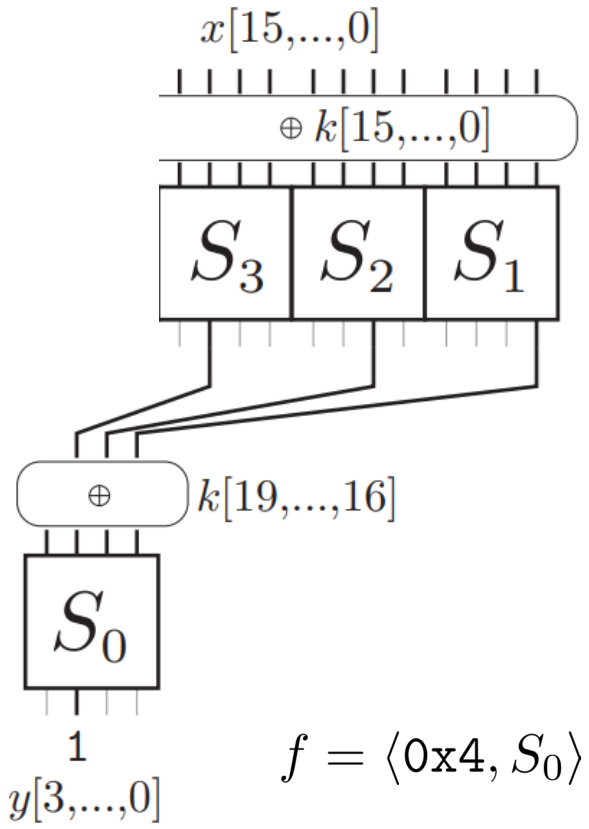
$f = \langle \texttt{0x4}, S_0 \rangle$

Understanding puncturing

Example, PRESENT-like cipher.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
f	1	-1	1	1	-1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1
$\hat{f}$	0	4	0	4	4	0	-4	8	4	0	-4	-8	0	4	0	4
$\hat{g}$	0	4	0	4	4	0	-4	8	0	0	0	0	0	0	0	0
g	1	-1	0	1	0	0	0	-1	1	-1	0	1	0	0	0	-1

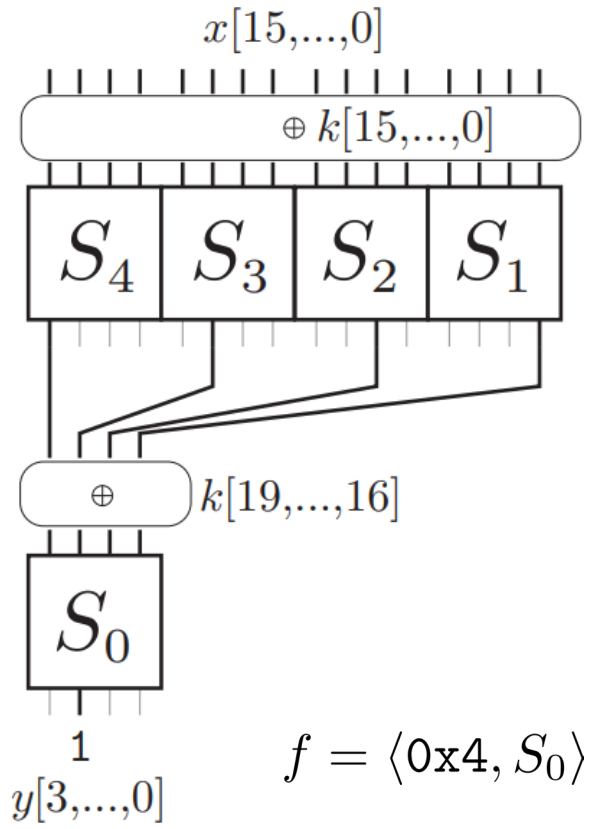
With $\rho^2 = 2^{-1}$, we can exclude 5-bit guess.



Understanding puncturing

Example, PRESENT-like cipher.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
f	1	-1	1	1	-1	1	-1	1	-1	-1	-1	1	1	-1	-1	-1
$\hat{f}$	0	4	0	4	4	0	-4	8	4	0	-4	-8	0	4	0	4
$\hat{g}$	0	4	0	4	4	0	-4	8	4	0	-4	-8	0	4	0	0
g	3/4	-3/4	5/4	3/4	-3/4	3/4	3/4	-3/4	5/4	-5/4	-5/4	5/4	3/4	-3/4	-3/4	-5/4



$f = \langle 0\text{x}4, S_0 \rangle$

With $\rho^2 = 1 - 2^{-4}$, the dimension of the key-recovery map decreases to 12.
Efficient key-recovery using the FWHT.

Apply puncturing to ARX…?

It’s practically difficult… Why?

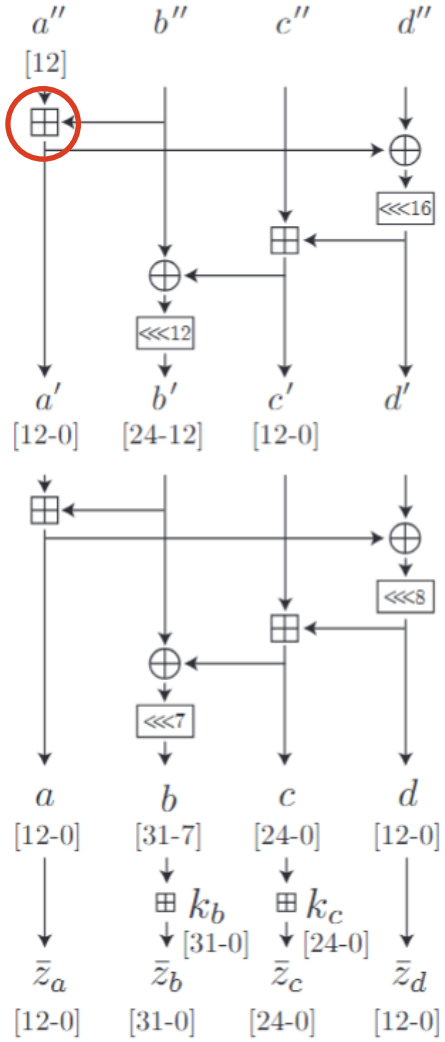
Example, Quarter rounds of ChaCha

- Assume that we want to evaluate a''[12].

$$(-1)^{a''[12]} = f(z_a[12-0], z_b[31-0], z_c[24-0], z_d[12-0], k_b[31-0], k_c[24-0]).$$

  - It involves 83-bit output and 57-bit key.

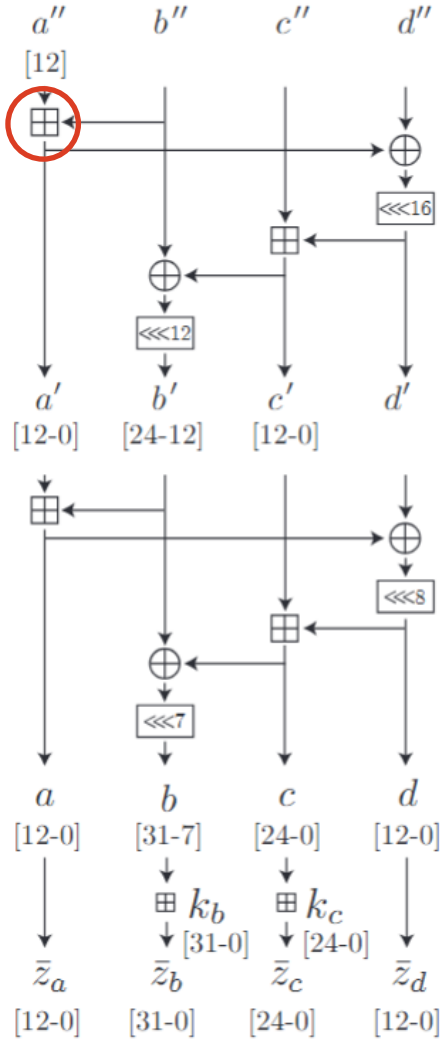
- Even if we apply the puncturing to the first modular addition (red circle), we can't exclude key bits from the key-recovery map.



Example, Quarter rounds of ChaCha

Useful observation

- Each Walsh coefficient is the correlation of a linear approximation, and it can be computed as the sum of the (signed) correlations of all linear trails in the approximation's linear hull.
- We enumerate many linear trails to recover Walsh spectrum coefficients.

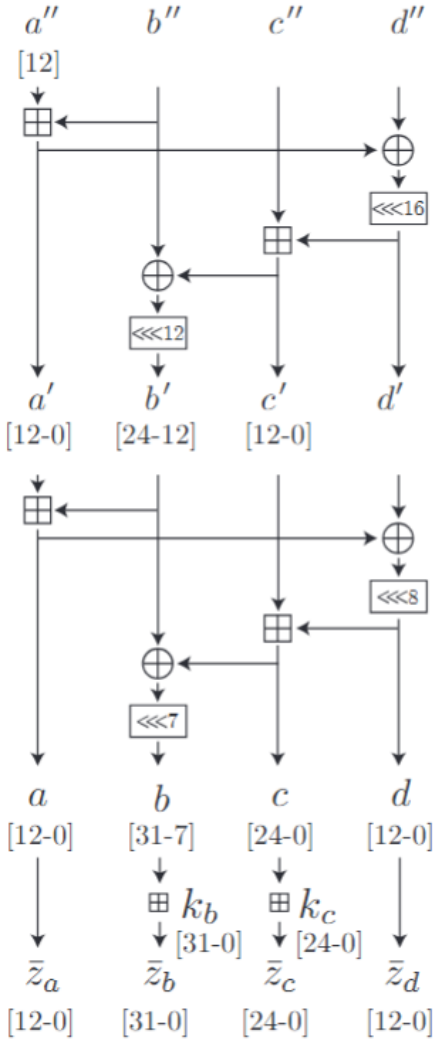


Trail enumeration puncturing

Understanding trail enumeration

1st step, target linear mask.

(00001000, 00000000, 00000000, 00000000) 1



Understanding trail enumeration

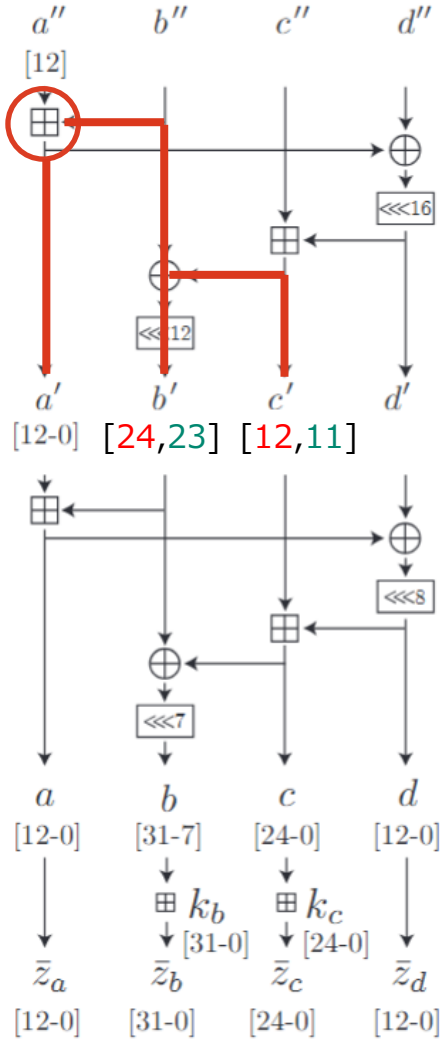
1st step, target linear mask.

(00001000, 00000000, 00000000, 00000000) 1

2nd step, evaluate linear transition of the modular addition.

(00001800, 01000000, 00001000, 00000000) 2^{-1}
(00001400, 01000000, 00001000, 00000000) 2^{-2}
…
(00001001, 01000000, 00001000, 00000000) 2^{-12}
(00001000, 01000000, 00001000, 00000000) 2^{-12}

(00001000, 01800000, 00001800, 00000000) 2^{-1}
(00001C00, 01800000, 00001800, 00000000) 2^{-2}
…
(00001801, 01800000, 00001800, 00000000) 2^{-12}
(00001800, 01800000, 00001800, 00000000) 2^{-12}



There are 26 coefficients.

Understanding trail enumeration

1st step, target linear mask.

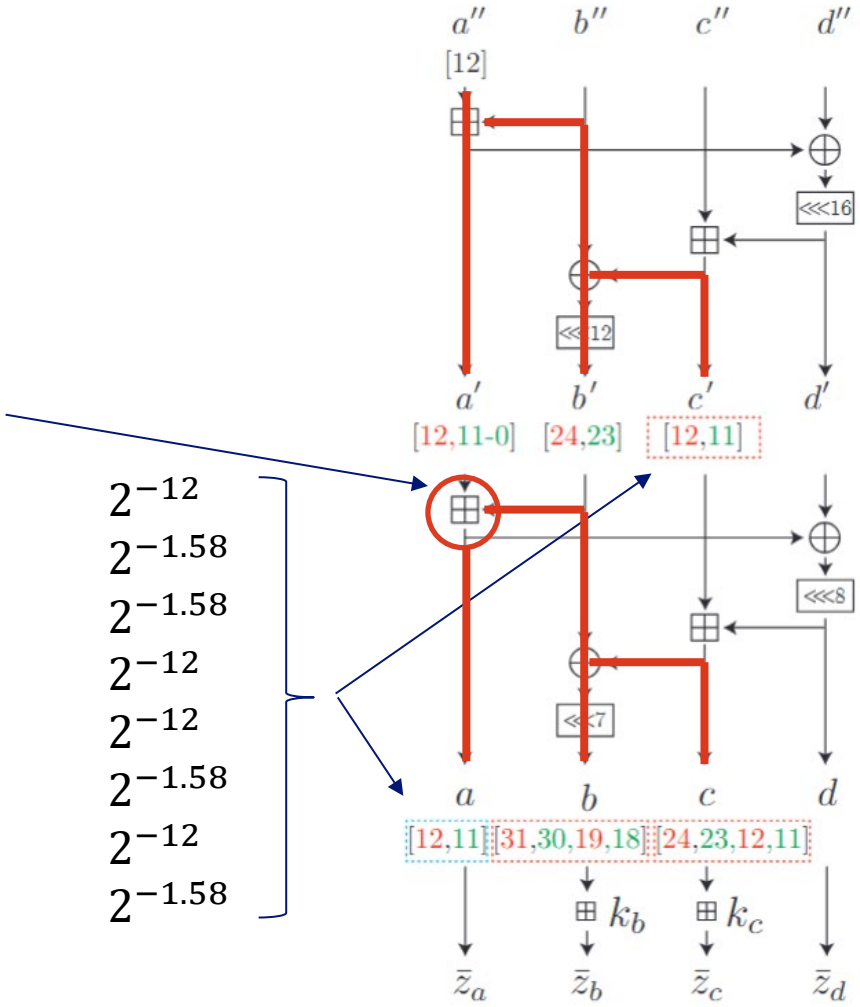
(00001000, 00000000, 00000000, 00000000) 1

2nd step, evaluate linear transition of the modular addition.

26 coefficients

3rd step, evaluate linear transition of the next modular addition.

(00001000, C0080000, 01801000, 00000000, 00001800)
(00001000, C00C0000, 01801800, 00000000, 00001800)
(00001800, C00C0000, 01001800, 00000000, 00001000)
(00001800, C0080000, 01001000, 00000000, 00001000)
(00001800, C00C0000, 01801800, 00000000, 00001800)
(00001800, C0080000, 01801000, 00000000, 00001800)
(00001000, C00C0000, 01001800, 00000000, 00001000)
(00001000, C0080000, 01001000, 00000000, 00001000)



There are 8 coefficients.

Understanding trail enumeration

1st step, target linear mask.

(00001000, 00000000, 00000000, 00000000) 1

2nd step, evaluate linear transition of the modular addition.

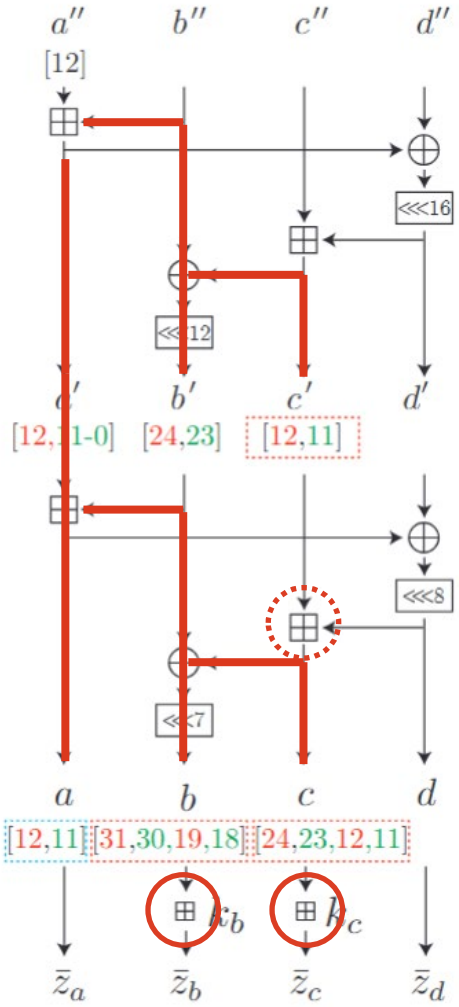
26 coefficients

3rd step, evaluate linear transition of the next modular addition.

8 coefficients

4th step, evaluate linear transition of the last key addition.

We guess Kb[31,30,19,18] and kc[24,23,12,11].
We use 2 bits for each keystream branch.
768 coefficients. 2^{10} dimension. Puncturing correlation $2^{-6.17}$.



Understanding trail enumeration

1st step, target linear mask.

(00001000, 00000000, 00000000, 00000000) 1

2nd step, evaluate linear transition of the modular addition.

26 coefficients

3rd step, evaluate linear transition of the next modular addition.

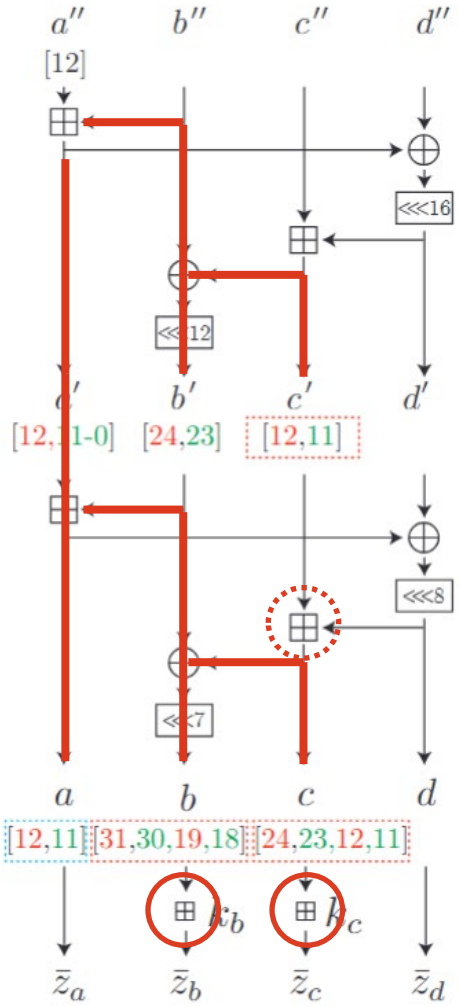
8 coefficients

4th step, evaluate linear transition of the last key addition.

We guess Kb[31,30,19,18] and kc[24,23,12,11].
We use 2 bits for each keystream branch.
768 coefficients. 2^{10} dimension. Puncturing correlation $2^{-6.17}$.

Obtain pseudoboolean function, g

Apply the Fast Walsh Transform (FWT), whose cost is 10×2^{10}.



What is different from PNBs?

PNBs

- Experimental
- Each output of the approximation is bool.
- Correlation depends on the key heavily (there are strong/weak keys).

Puncturing

- Theoretical
- Each output of the approximation is real value.
- Correlation is the key-average one (conceptually, it's optimal approximation).

Key recovery attack on ChaCha using Puncturing

Attack against ChaCha6

4.5-round DL distinguisher (BLT20)

$$(\Delta_{12}^{0}[6]) \xrightarrow{1R} (\Delta_{0}^{1}[2], \Delta_{4}^{1}[29,17,9,5], \Delta_{8}^{1}[30,22,10], \Delta_{12}^{1}[30,10]) \qquad p = 2^{-5}$$

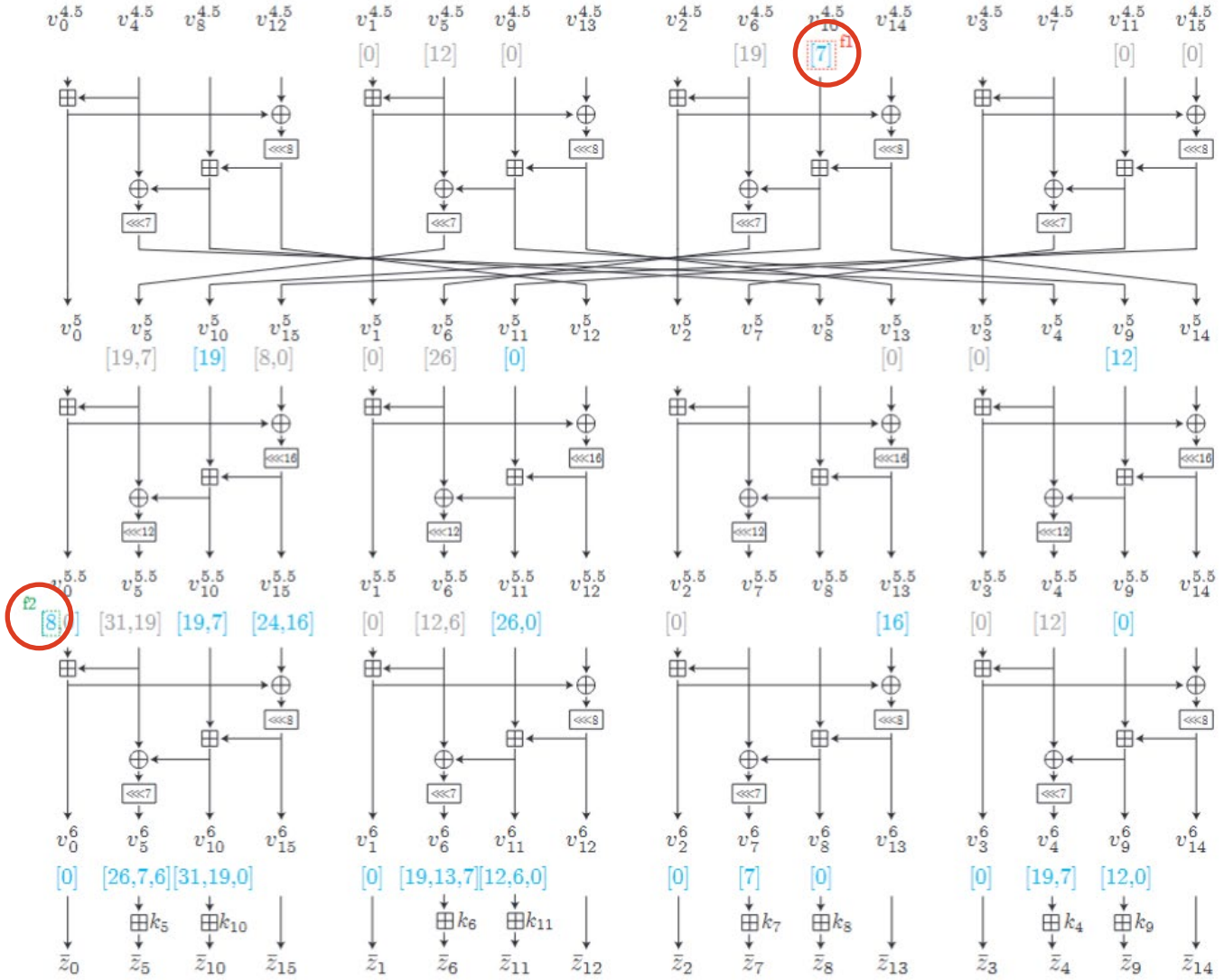
$$(\Delta_{0}^{1}[2], \Delta_{4}^{1}[29,17,9,5], \Delta_{8}^{1}[30,22,10], \Delta_{12}^{1}[30,10]) \xrightarrow{2.5R} (\Gamma_{1}^{3.5}[0]) \qquad c = 2^{-8.3}$$

$$(\Gamma_{1}^{3.5}[0]) \xrightarrow{1R} (\Gamma_{1}^{4.5}[0], \Gamma_{5}^{4.5}[12], \Gamma_{6}^{4.5}[19], \Gamma_{9}^{4.5}[0], \Gamma_{10}^{4.5}[7], \Gamma_{11}^{4.5}[0], \Gamma_{15}^{4.5}[0]) \qquad c = 1$$

Attack against ChaCha6

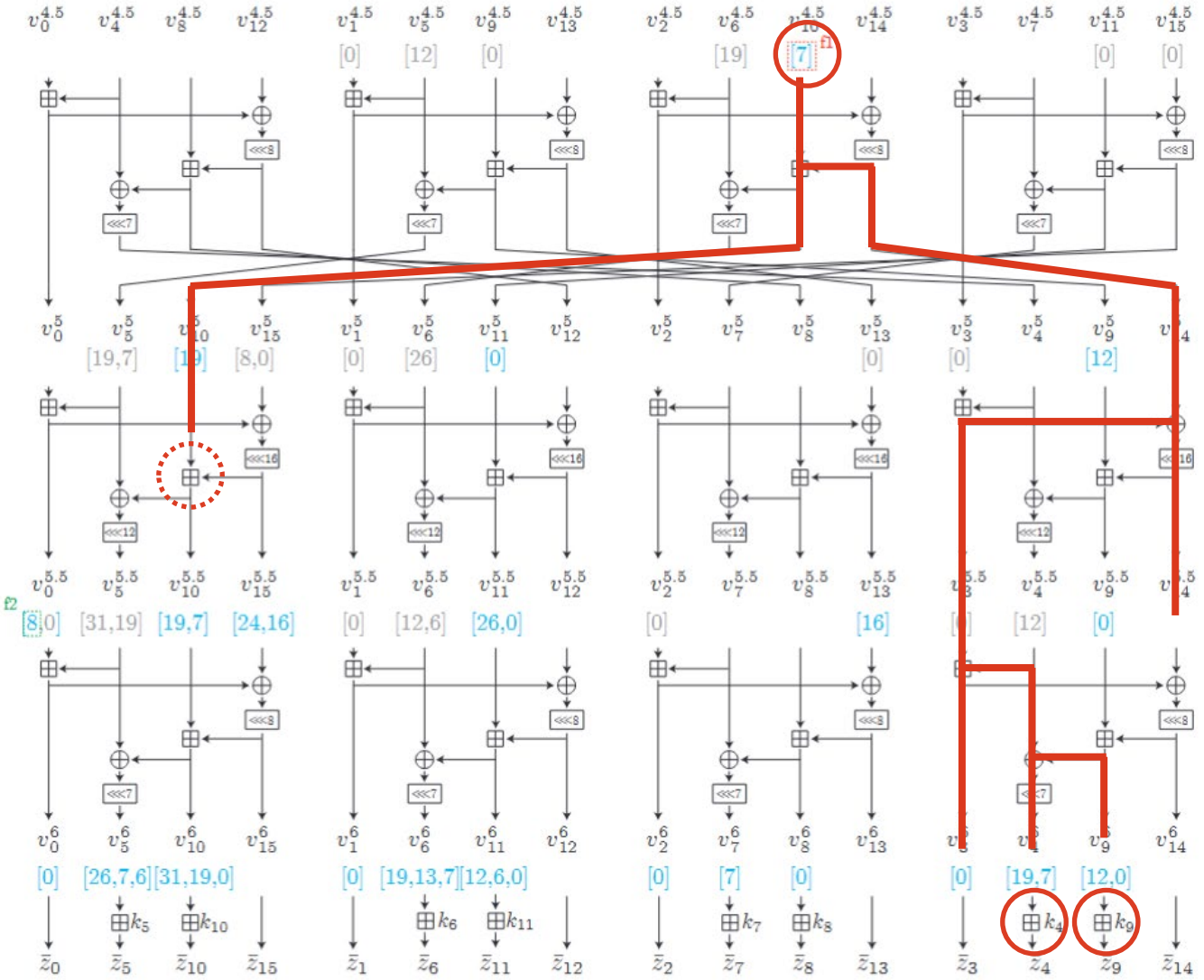
Only two target bits, $v_{10}^{4.5}[7]$ and $v_{0}^{5.5}[8]$, involves more than one modular addition.

The others (18) involve at most one modular addition.



Attack against ChaCha6

f1, $v_{10}^{4.5}[7]$



Attack against ChaCha6

f1, $v_{10}^{4.5}[7]$

4-bit guess

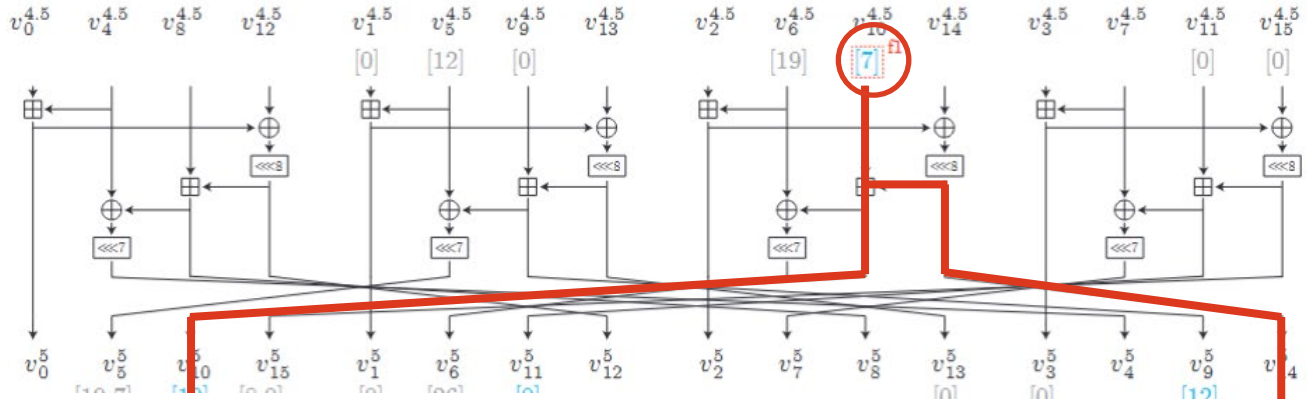
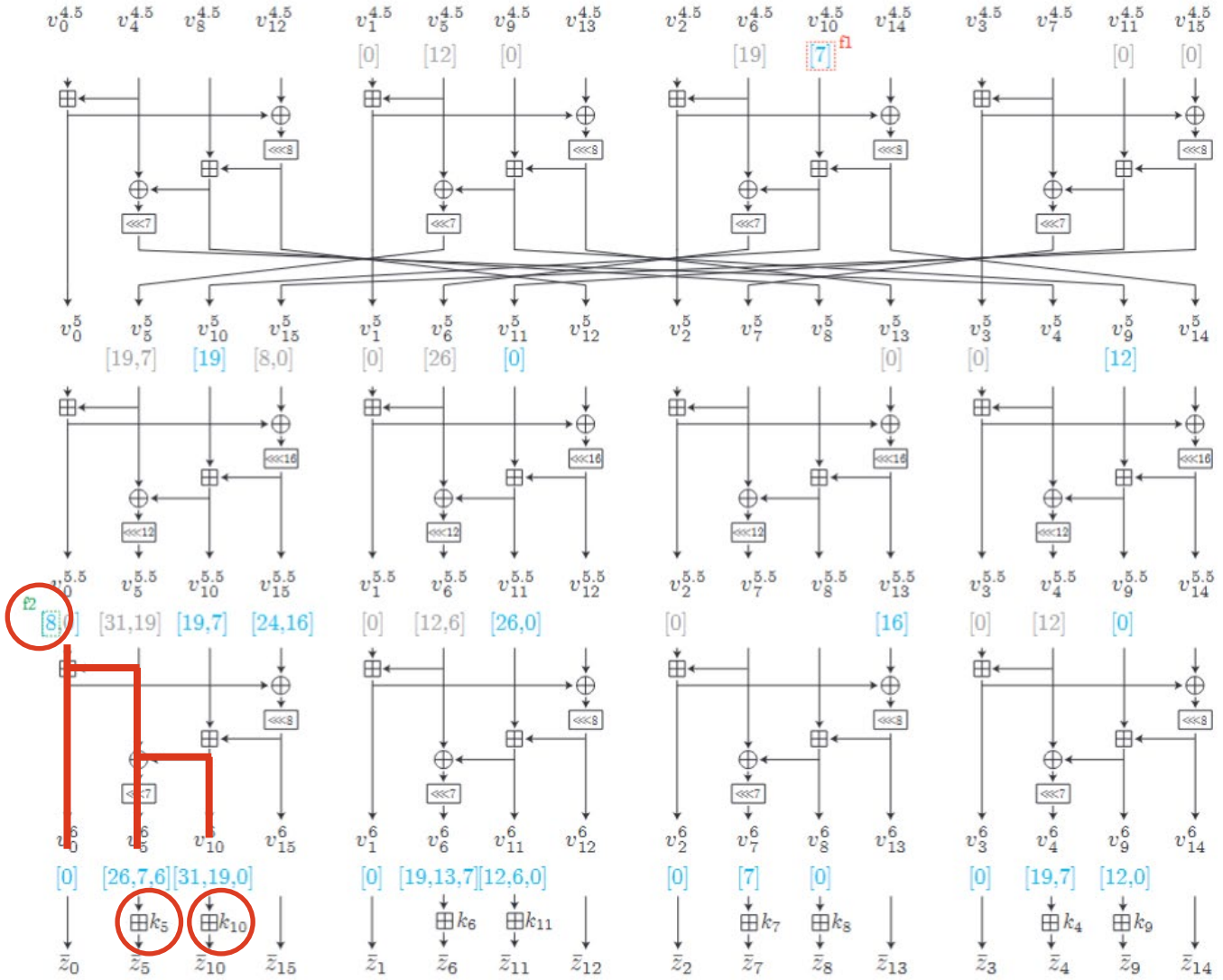


Table 4: Trail enumeration for $f_1 = (-1)^{v_{10}^{4.5}[7]}$

round	active	#coeffs	ρ^2
5	$v_{10}^{5}[7,6\text{-}0]$, $v_{14}^{5}[7,6\text{-}2]$	158	$2^{-0.029}$
5.5	$v_{10}^{5}[7,6\text{-}0]$, $v_{3}^{5.5}[7,6\text{-}2]$, $v_{14}^{5.5}[23,22\text{-}18]$	158	$2^{-0.029}$
6	$v_{10}^{5}[7,6\text{-}0]$, $v_{14}^{5.5}[23,22\text{-}18]$, $v_{3}^{6}[7,6\text{-}2]$, $v_{4}^{6}[14,13\text{-}9]$, $v_{9}^{6}[7,6\text{-}2]$	24248	$2^{-0.15}$
end	$\bar{z}'_{14}[23,22\text{-}18]$, $\bar{z}_3[7,6\text{-}2]$, $v_{10}^{5}[7,6\text{-}0] \leftarrow \{\bar{z}''_{10}[7,6\text{-}2], k_{10}[7,6,5]\}$, $v_{4}^{6}[14,13\text{-}9] \leftarrow \{\bar{z}_4[14,13\text{-}9], k_4[14,13]\}$, $v_{9}^{6}[7,6\text{-}2] \leftarrow \{\bar{z}_9[7,6\text{-}2], k_9[7,6]\}$	3699840	$2^{-2.31}$

Attack against ChaCha6

f2, $v_0^{5.5}[8]$



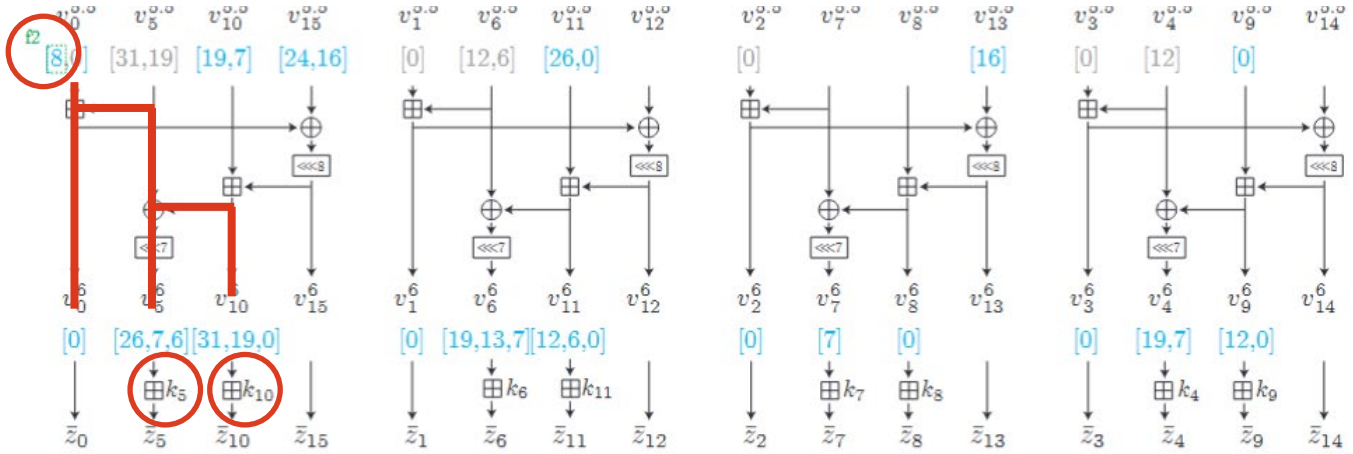
Attack against ChaCha6

f2, $v_0^{5.5}[8]$

4-bit guess

Table 3: Trail enumeration for $f_2 = (-1)^{v_0^{5.5}[8]}$

round	active	#coeffs	ρ^2
6	v_0^6[8,7-3], v_5^6[15,14-10], v_{10}^6[8,7-3]	94	$2^{-0.045}$
end	$\bar{z}_0$[8,7-3], v_5^6[15,14-10] $\leftarrow \{\bar{z}_5$[15,14-10], k_5[15,14]$\}$, v_{10}^6[8,7-3] $\leftarrow \{\bar{z}_{10}$[8,7-3], k_{10}[8,7-5]$\}$	18416	$2^{-1.23}$



Simple procedure

- Parameter
  - Puncturing correlation for f1 and f2, $2^{-2.31} \times 2^{-1.23} = 2^{-3.54}$.
  - There are 18 functions consisting of a modular addition.
    - We guess 2-bit key for each function.
    - Use partitioning, $\left(\frac{3}{4}\right)^{18} \approx 2^{-7.47}$.
  - Total correlation, $2^{-3.54} \times 2^{-7.47} = 2^{-11.01}$.
  - Guess key bits (34 bits in total).

$N = N' \times 2^{11.01 \times 2} = 2^{45.02}$ pairs

$2N \times 2^{-7.17 \times 2} \times 2^{34} = 2^{65.08}$ time

$2^{51.02}$ ks blocks

$2^{70.08}$ time

2^5 repats

Optimization using distillation table

Distillation table [Matsui, CRYPTO94]

- Instead of guessing the key for each data, we store involved bits only and generate the distillation table.
- The impact. $N \times 2^k \Rightarrow N + 2^{b+k}$
  - If $N > 2^b$, using the distillation table is useful.
- Two-step procedure improves the attack.

$N = N' \times 2^{11.01 \times 2} = 2^{45.02}$ pairs

$2N \times 2^{-4.15 \times 2} \times 2^{18} + 2^{54.98} = 2^{56.40}$ time

2^5 repats

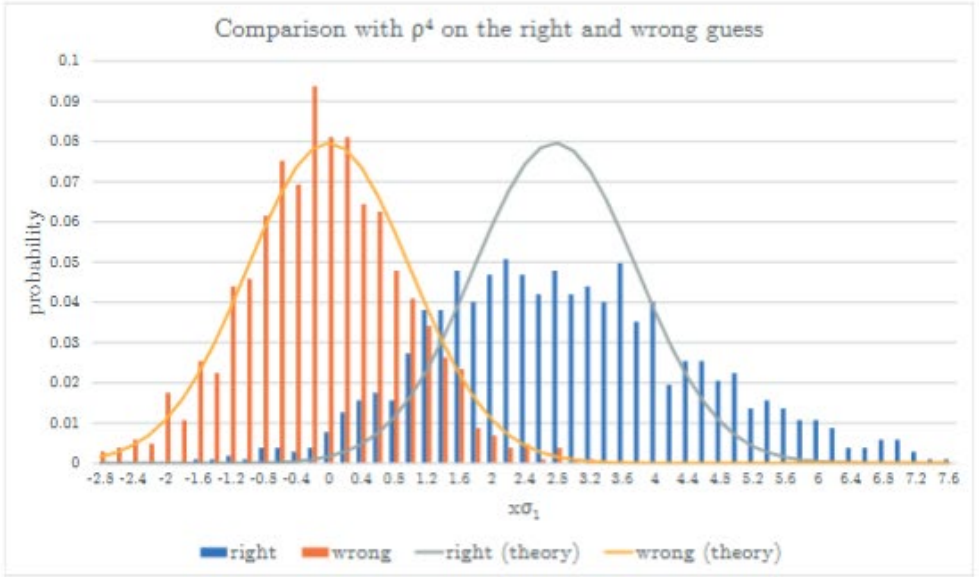
$2^{51.02}$ ks blocks

$2^{61.40}$ time

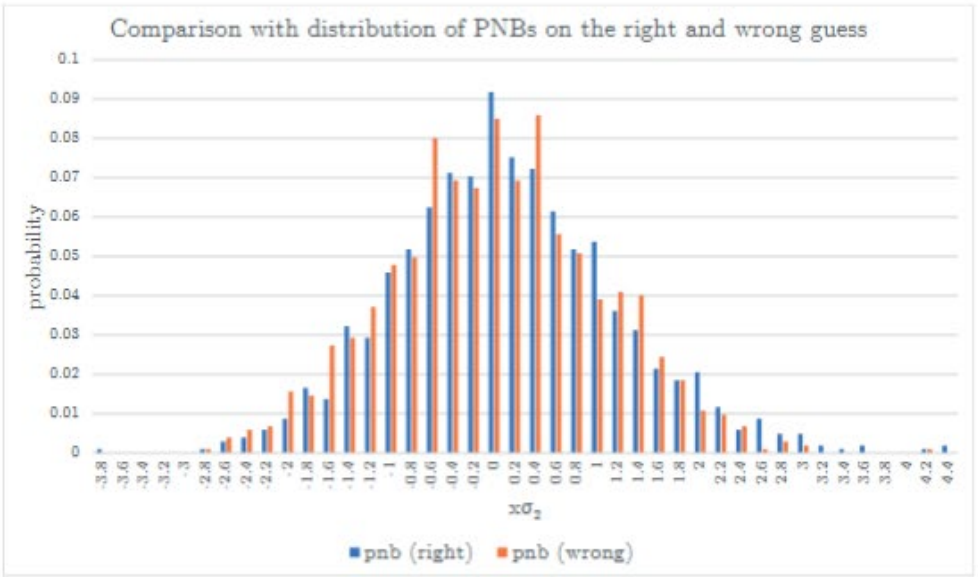
Experiments on 6-round attack

Comparison of backward correlation

- 2^{12} samples, 1000 random keys, 5-bit guess for f1 and f2.



(a) Experiments for ρ^4.



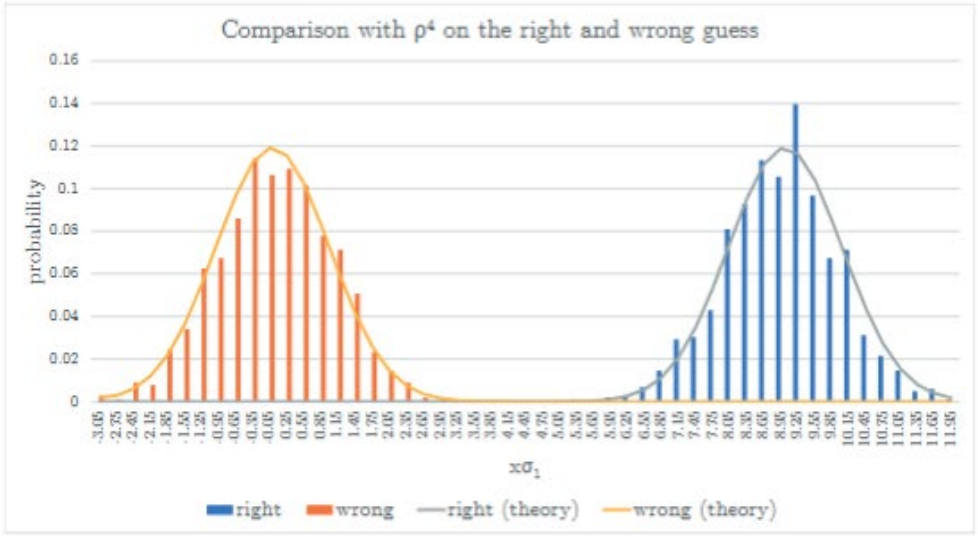
(b) Experiments for PNBs.

We have distinguishable distribution in puncturing, but don't have in PNBs.

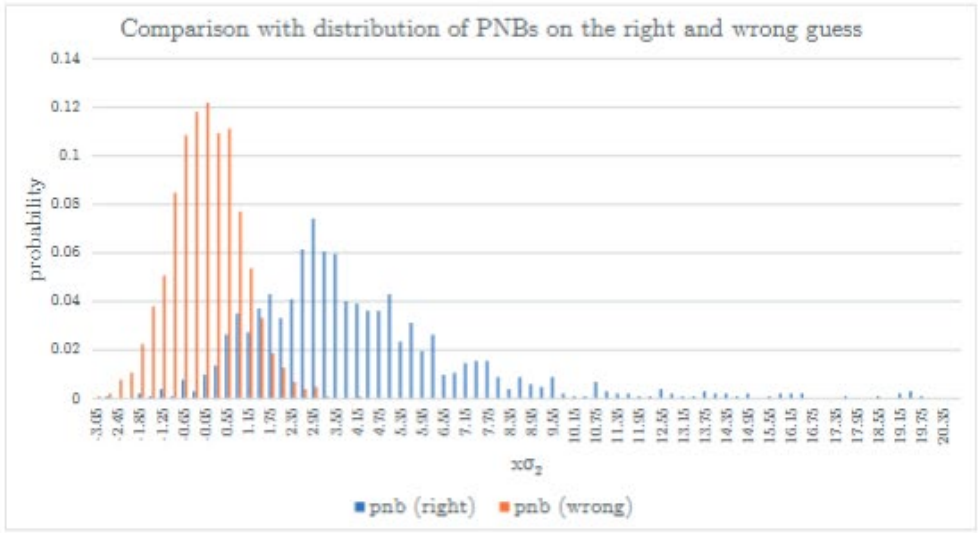
Experiments on 6-round attack

Comparison of backward correlation

- 2^{12} samples, 1000 random keys, 13-bit guess for f1 and f2.



(a) Experiments for ρ^4.



(b) Experiments for PNBs.

PNBs also distinguishable, but clearly, puncturing is better.

Attack on 7-round ChaCha

Attack against ChaCha7

5-round DL distinguisher (BGG+23, XXTQ24)

$$(\Delta^0_{15}[29], \Delta^0_{15}[19]) \xrightarrow{1R} (\Delta^1_3[25,5], \Delta^1_7[28,12], \Delta^1_{11}[25,21], \Delta^1_{15}[21,13]) \qquad p = 2^{-7}$$

$$(\Delta^1_3[25,5], \Delta^1_7[28,12], \Delta^1_{11}[25,21], \Delta^1_{15}[21,13]) \xrightarrow{4R} (\Gamma^5_2[0], \Gamma^5_6[19,7], \Gamma^5_{10}[12], \Gamma^5_{14}[0]) \qquad c = 2^{-34.15}$$

Linear hull

$$c = 2^{-32.2}$$

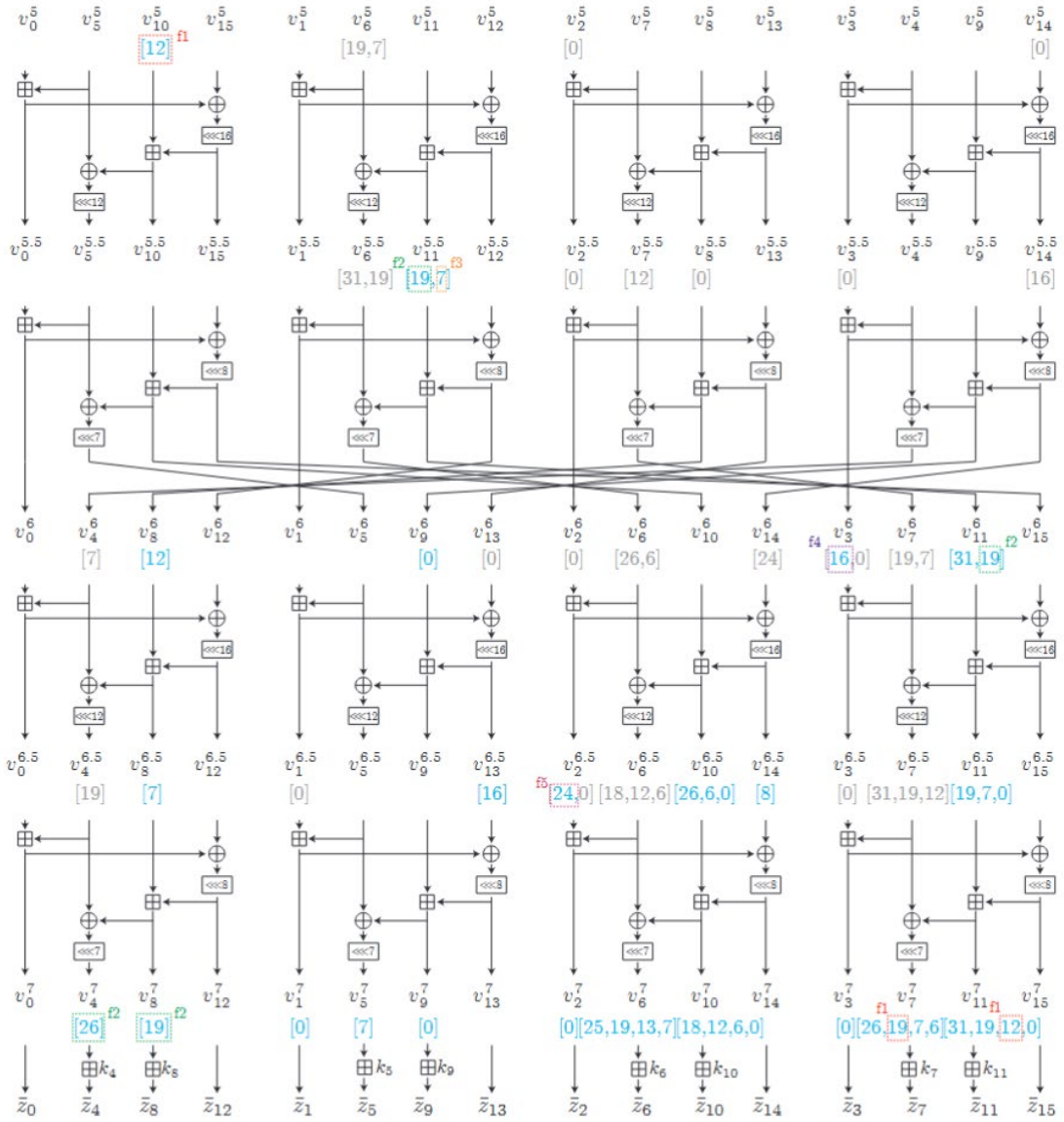
We have two types of the attack.

- The first uses 5-round DL with autocorrelation $2^{-39.2}$.
- Another uses the same trick using the friend pairs.

Attack against ChaCha7

There are five target bits, which involves more than one modular addition.

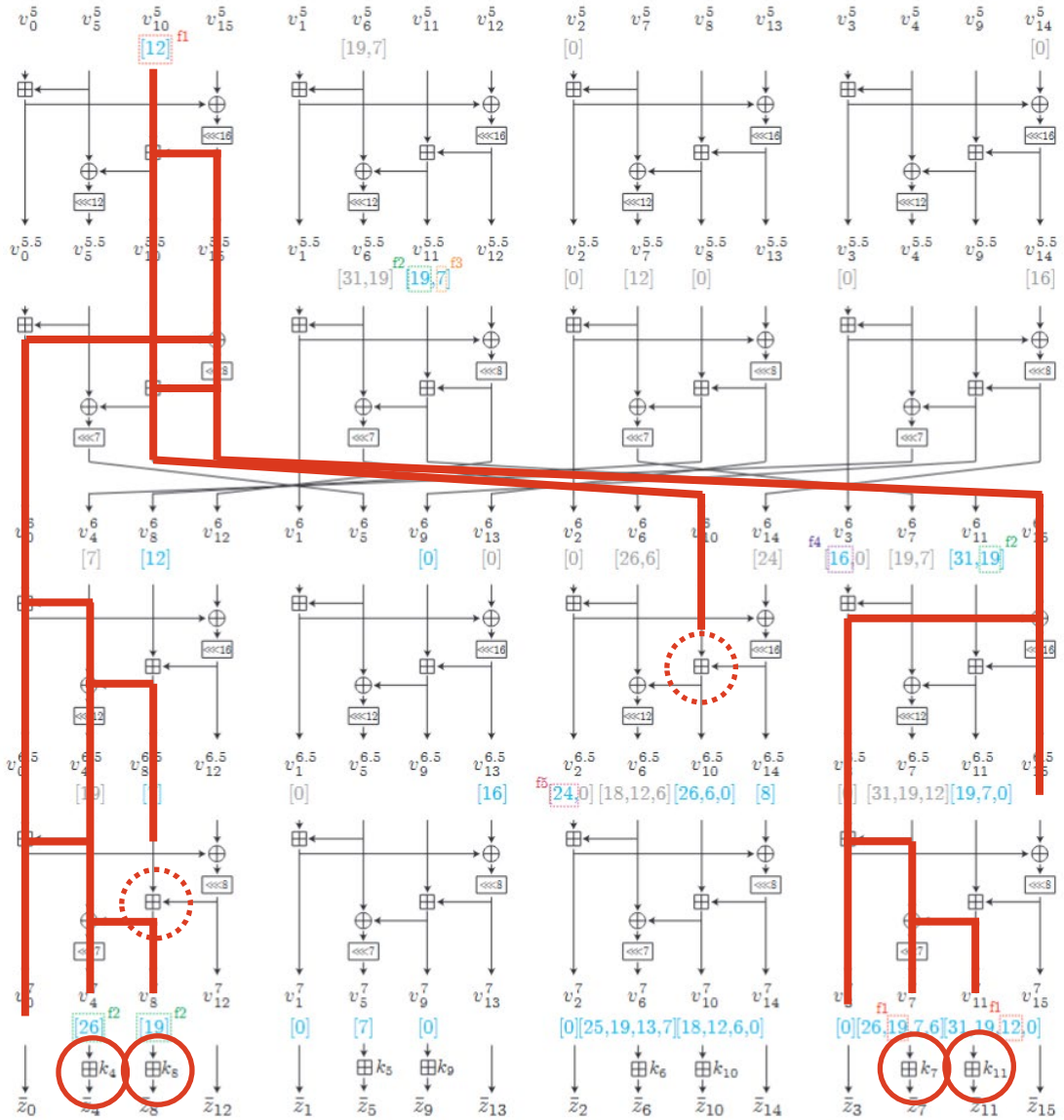
The others (19) involve at most one modular addition.



Attack against ChaCha7

There are five target bits, which involves more than one modular addition.

The others (19) involve at most one modular addition.



Attack against ChaCha7

Table 5: Punctured functions for the 7-round attack.

Target	Active	#coeffs	ρ^2
f_1	$\bar{z}_0[12,11\text{-}9]$, $\bar{z}_3[20,19\text{-}17,12,11\text{-}9]$, $\bar{z}'_{15}[28,27\text{-}25,4,3\text{-}1]$, $v_4^7[31,30\text{-}28,19,18\text{-}16] \leftarrow \{\bar{z}_4[31,30\text{-}26,19,18\text{-}14], k_4[31,30,19,18]\}$, $v_7^7[27,26\text{-}24] \leftarrow \{\bar{z}_7[27,26\text{-}22], k_7[27,26,22]\}$, $v_8^7[24,23\text{-}21,12,11\text{-}9] \leftarrow \{\bar{z}_8[24,23\text{-}19,12,11\text{-}7], k_8[24,23,12,11,10]\}$, $v_{11}^7[20,19\text{-}17] \leftarrow \{\bar{z}_{11}[20,19\text{-}15], k_{11}[20,19\text{-}17,15]\}$, $v_8^{6.5}[12,11\text{-}9] \leftarrow \{\bar{z}'_8[12,11\text{-}7], k'_8[12,11,10]\}$, $v_{10}^6[12,11\text{-}9] \leftarrow \{\bar{z}''_{10}[12,11\text{-}7], k''_{10}[12,11]\}$	$\leq 2^{55+14}$	$2^{-7.14}$
f_2	$\bar{z}_0[19,18\text{-}14]$, $\bar{z}'_{12}[3,2\text{-}0,31,30]$, $v_{11}^6[18\text{-}14] \leftarrow \{\bar{z}''_{11}[18\text{-}13], k''_{11}[18,17,15,14]\}$	1740	$2^{-2.39}$
f_3	$\bar{z}_0[7,6\text{-}2]$, $\bar{z}'_{12}[23,22\text{-}18]$, $v_8^7[7,6\text{-}2] \leftarrow \{\bar{z}_8[7,6\text{-}2], k_8[7,6]\}$, $v_4^7[14,13\text{-}9] \leftarrow \{\bar{z}_4[14,13\text{-}9], k_4[14,13]\}$, $v_{11}^6[7,6\text{-}2] \leftarrow \{\bar{z}''_{11}[7,6\text{-}2], k''_{11}[7,6]\}$	1570720	$2^{-2.57}$
f_4	$\bar{z}_3[16,15\text{-}11]$, $v_7^7[23,22\text{-}18,3,2\text{-}0] \leftarrow \{\bar{z}_7[23,22\text{-}18,3,2\text{-}0], k_7[23,22,3,2]\}$, $v_{11}^7[28,27\text{-}25,16,15\text{-}11] \leftarrow \{\bar{z}_{11}[28,27\text{-}23,16,15\text{-}11], k_{11}[28,27,16,15,14]\}$, $v_{11}^{6.5}[16,15\text{-}13] \leftarrow \{\bar{z}'_{11}[16,15\text{-}11], k_{11}[16,15,14]\}$	$\leq 2^{28+7}$	$2^{-3.07}$
f_5	$\bar{z}_2[24,23\text{-}19]$, $v_6^7[31,30\text{-}26] \leftarrow \{\bar{z}_6[31,30\text{-}26], k_6[31,30]\}$, $v_{10}^7[24,23\text{-}19] \leftarrow \{\bar{z}_{10}[24,23\text{-}19], k_{10}[24,23]\}$	2144	$2^{-1.74}$

Summary of Results

Round	Data	Time	Note	Ref
6	2^73.7	2^75.5	PNBs w/ syncopation	Wang et al., CRYPTO, 2023
	2^41.6	2^71.0	PNBs w/ linear decomposition	Dey, IEEE-IT, 2024
	2^51,0	**2^61.4**		Ours
	2^55.7	**2^57.4**		Ours
7	2^102.6	2^189.7	DL hull and PNBs	Xu et al., ToSC, 2024
	2^127.7	**2^148.2**		Ours
	2^102.9	**2^154.2**		Ours
7.5	2^32.6	2^255.2	PNBs w/ linear decomposition	Dey, IEEE-IT, 2024
	2^127.1	**2^250.2**		Ours

Summary

Summary

- A new tool to analyze ChaCha.
  - No more PNBs. Fully theoretical analysis is possible!!
- Significant improvement of ChaCha
  - 6 rounds, 2^{71} time※ → $2^{57.4}$ time
  - 7 rounds, $2^{189.7}$ time → $2^{154.2}$ time
  - 7.5 rounds, $2^{255.2}$ time※ → $2^{250.2}$ time

Each cost is one table lookup.
We regard the cost is equivalent with one encryption.

- Next step (coming soon)
  - Application to Salsa.
  - Additional technique to improve the efficiency.

Next step and Open problem

- Our next step (coming soon)
  - Application to Salsa.
  - Additional technique to improve the efficiency.
- Open problem (inspired by the motivation of SKCAM).
  - Probably, everyone loves **differential** than **linear**...?
    - Linear key recovery might be more complicated.
      Walsh spectrum, FWHT, and now, puncturing.
  - Automation (including the puncturing) is not easy...?
    - So far, choosing parameter is my heuristic.
    - There are no automatic tool.