Beyond-Full-Round Integral Distinguisher of NIST Lightweight Cryptography Competition Finalist TinyJAMBU

Akram Khalesi and Zahra Ahmadian

Department of Electrical Engineering, Shahid Beheshti University, Tehran, Iran,
a_khalesi@sbu.ac.ir
z_ahmadian@sbu.ac.ir

Abstract. TinyJAMBU is one of the ten finalists of the NIST lightweight cryptography competition, announced in March 2021. It proposes a lightweight authenticated encryption scheme based on a lightweight 128-bit keyed permutation. TinyJAMBU supports three key lengths 128, 192, and 256 denoted by TinyJambu-128, TinyJambu-192, and TinyJambu-256, respectively. The scheme as well as the permutation is well studied by the designers and third parties. The most relevant work to ours is the full-round zero-sum distinguisher under the known-key setting assumption published at Indocrypt 2022. In this work, we show that even without the known-key setting assumption, there are integral distinguishers not only for full-round versions of the permutations of TinyJambu-128 and TinyJambu-192 but also for round-increased versions of them up to 1273 rounds.

Keywords: TinyJAMBU · Lightweight Cryptography · Division Property · Integral Distinguisher · MILP

1 Introduction

NIST has been working to standardize lightweight cryptographic algorithms since August 2018 [1]. The process began with a call for lightweight Authenticated Encryption (AE) algorithms and lightweight hash functions, which resulted in 57 submissions. From these, 56 algorithms were selected as first-round candidates, and 32 passed to the second round in August 2019. In March 2021, NIST concluded the second round by introducing 10 finalists, which are ASCON, Elephant, GIFT-COFB, Grain128-AEAD, ISAP, Photon-Beetle, Romulus, Sparkle, TinyJambu, and Xoodyak. After careful evaluation, NIST has selected ASCON as the winner, which will be published as NIST's lightweight cryptography standard later in 2023. This standardization process is crucial for ensuring the security of small devices and platforms, such as those used in the Internet of Things.

TinyJAMBU is a lightweight AE with a structure similar to the duplex sponge constructions [2]. It uses a keyed permutation based on a nonlinear feedback shift register (NLFSR) with a 128-bit internal state. The only nonlinear component of TinyJAMBU permutation is a two-input NAND gate. It supports three key lengths: 128, 192, and 256 bits.

The division property was first introduced by Todo at Eurocrypt 2015 [3] as a tool for studying the integral property of block ciphers. Todo et al. later proposed the bit-based version of the division property, as well as the three-subset version of the bit-based division property at FSE 2016 [4]. The bit-based version is mainly used after its introduction, so it is often shortly referred to as the "division property", briefly. We will also refer to it as such in this paper.

Xiang et al. proposed a method for automating the search of integral distinguishers based on the division property [5], making it possible to use the division property for cube attacks [6]. Both versions of the division property, i.e. conventional division property and three-subset division property, have been well studied and automated [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. As the outcome of these extensive researches, the division property has turned to be not only a powerful technique to find integral distinguishers but also an efficient method for cube attacks and in the assessment of the Algebraic Normal Form (ANF) of boolean functions.

Dutta et al. conducted a security evaluation of TinyJAMBU in three contexts [20]. Firstly, a cube attack in the weak-key setting is proposed where at least 2^{108} key values are identified for which P_r, for r up to 476, can be distinguished from a random permutation. Secondly, they investigated the exact degree of feedback polynomial in the nonce variables and reported that this degree for P_{381} is 32. Finally, they proposed a key recovery attack on P_{440}. It deserves to note that their analyzes do not rely on inadmissible assumptions such as nonce-reuse.

At Indocrypt 2022, Dunkelman et al. reported full-round zero-sum distinguishers for the permutation of TinyJAMBU under the known-key setting assumption [21]. They have deployed the monomial prediction technique proposed by [14] to evaluate the algebraic degree of the permutation in the forward and backward directions. They also utilized the inside-out approach that imposes the known-key setting assumption. Using a trade-off on data/time complexity to the number of balanced bits, the best proposed zero-sum distinguisher on the full round of the permutation is with data/time complexity 2^{16} and $r_1 = 480$ and $r_2 = 544$ rounds in the inside-out approach. One can refer to [21] for more details.

Our Contributions. In this paper, we propose a security evaluation of TinyJAMBU with regard to integral attack by proposing new distinguishers on full-round and even beyond-full-round permutation of TinyJAMBU without known-key setting assumption. The results are summarized in Table 1.

We have utilized the conventional division property for finding the integral distinguishers. We observed that although the conventional division property might miss some balanced properties, and hence returns a sub-optimum solution, it can be applied on more rounds of a permutation due to its lower complexity compared to the no-missing methods. The no missing methods such as "Modeling for three-subset division property without unknown subset" [12] or its counterpart "the monomial prediction technique" [14] lead to more accurate results, but their complexity is prohibitive in many cases.

Outline. The paper is organized as follows: we first introduce the notations used in the paper as well as the division property in Section 2. A brief introduction to TinyJAMBU is given in Section 3. In Section 4, the results are reported and finally, we conclude the paper in Section 5.

2 Preliminaries

In this section, we first review the notations used in the paper. Then an introduction to the division property for integral cryptanalysis is given.

2.1 Notations

The list of notations used in this document is summarized in Tab. 2.

Table 1: Comparison of integral cryptanalysis of TinyJAMBU

Permutation	Data	Model	Attack	Target	Ref.
P_{1273}	2^{127}	Secret Key	Integral	Distinguishing	section 4.1
P_{1152}	2^{127}	Secret Key	Integral	Distinguishing	section 4.2
P_{1024}	2^{126}	Secret Key	Integral	Distinguishing	section 4.3
P_{1024}	2^{127}	Secret Key	Integral	Distinguishing	section 4.3
P_{544}	2^{23}	Secret Key	Zero-sum	Distinguishing	[21]
P_{480}	2^{16}	Secret Key	Zero-sum	Distinguishing	[21]
P_{476}	2^{16}	Weak Key	Cube	Distinguishing	[20]
P_{440}	2^{31}	Secret Key	Cube	Key Recovery	[20]
P_{438}	2^{32}	Secret Key	Cube	Distinguishing	[22]
P_{438}	2^{32}	Secret Key	Integral	Distinguishing	section 4.4
P_{428}	2^{32}	Secret Key	Cube	Key Recovery	[22]
P_{1024}	2^{16}	Known Key	Zero-sum	Distinguishing	[21]
P_{1152}	2^{23}	Known Key	Zero-sum	Distinguishing	[21]

Table 2: List of Notations used in the paper

Notation	Meaning
$\oplus$	Bit-wise *xor*
$\&$	Bit-wise *and*
$\neg$	Bit-wise *negation*
$\mathbb{F}_2$	Binary finite field
$\boldsymbol{a} = [a_0, \ldots, a_{n-1}] \in \mathbb{F}_2^n$	n-bit vector (a_i denotes the i-th bit of $\boldsymbol{a}$)
$\boldsymbol{e}_i$	The unit vector whose i-th element is 1
$Hw(\boldsymbol{a})$	Hamming weight of $\boldsymbol{a}$, i.e. $\sum_{i=0}^{n} a_i$
$\boldsymbol{k} \succeq \boldsymbol{k}'$	$k_i \geq k'_i$ for all $i \in \{0, \ldots, n-1\}$
$\boldsymbol{x}^{\boldsymbol{u}}$	$\prod_{i=0}^{n-1} {x_i}^{u_i}$

2.2 Division Property

Division property comes in two versions: conventional division property [3] and three-subset division property [4]. Definitions of these two versions are as follows:

Definition 1. Conventional Division Property[3]: *Let $\mathbb{X}$ be a multiset whose elements take value from F_2^n. The multiset $\mathbb{X}$ has the conventional bit-based division property $D_{\mathbb{K}}^{1^n}$ if it fulfills the following condition:*

$$\bigoplus_{x \in \mathbb{X}} \boldsymbol{x}^{\boldsymbol{u}} = \begin{cases} unknown, & \text{if there exists } \boldsymbol{k} \in \mathbb{K} \text{ s.t. } \boldsymbol{u} \succeq \boldsymbol{k}, \\ 0, & \text{otherwise} \end{cases}.$$

where $\mathbb{K}$ denotes a set of n-dimensional binary vectors.

Definition 2. Three-Subset Division Property[4]: *Let $\mathbb{X}$ be a multiset whose elements take value from F_2^n. The multiset $\mathbb{X}$ has the three-subset bit-based division property $D^{1^n}_{\mathbb{K},\mathbb{L}}$ if it fulfills the following condition:*

$$\bigoplus_{x\in\mathbb{X}} \boldsymbol{x}^{\boldsymbol{u}} = \begin{cases} unknown, & \text{if there exists } \boldsymbol{k}\in\mathbb{K} \text{ s.t. } \boldsymbol{u}\succeq\boldsymbol{k}, \\ 1, & \text{if there exists } \boldsymbol{l}\in\mathbb{L} \text{ s.t. } \boldsymbol{u}=\boldsymbol{l}, \\ 0, & otherwise \end{cases} .$$

where $\mathbb{K}$ and $\mathbb{L}$ denote sets of n-dimensional binary vectors.

The propagation rules of division property for different basic components in symmetric ciphers ("*copy*", "*and*", "*xor*" and "*SBox*") and their corresponding MILP models are given in [4] and [5], respectively.

3 Specifications of TinyJAMBU

TinyJAMBU family of lightweight authenticated encryption algorithms consists of three versions TinyJAMBU-128, TinyJAMBU-192, and TinyJAMBU-256. All three versions are the same in the nonce length (96 bits), authentication tag (64 bits), and the associated data as well as the plaintext lengths (both up to 2^{50} bytes). They differ in the key lengths and the permutation number of rounds. TinyJAMBU-n has key length of n-bit and the round number of the permutation in the initialization, encryption and finalization phases depends on the version/key length and is pointed out in sections 3.1, 3.3 and 3.4. The TinyJAMBU mode is depicted in Fig. 1. It consists of four phases:

- Initialization
- Processing the associated data
- Encryption
- Finalization

In all of these phases, different numbers of rounds of a 128-bit keyed permutation P_n as shown in Fig. 2 are used. Permutation P_n consists of n rounds and a 128-bit NFSR is used to update the state $\boldsymbol{s}$ with key $\boldsymbol{k}$ in the i^{th} round of P_n as follows:

$$\begin{aligned} &\text{StateUpdate}(\boldsymbol{s},\boldsymbol{k},i): \\ &\qquad \text{feedback} = s_0 \oplus s_{47} \oplus \neg(s_{70}\&s_{85}) \oplus s_{91} \oplus k_{i \bmod klen} \\ &\qquad \text{for } j \text{ from 0 to 126: } s_j = s_{j+1} \\ &\qquad s_{127} = \text{feedback} \end{aligned} \tag{1}$$

3.1 The Initialization Phase

The initialization phase consists of *key setup* and *nonce setup*. The numbers of rounds r of P_r in the key setup for key lengths 128, 192 and 256 are 1024, 1152, and 1280, respectively.
Key setup:

- Initialize the 128-bit state $\boldsymbol{s}$ with zeroes.
- Update $\boldsymbol{s}$ with P_r.

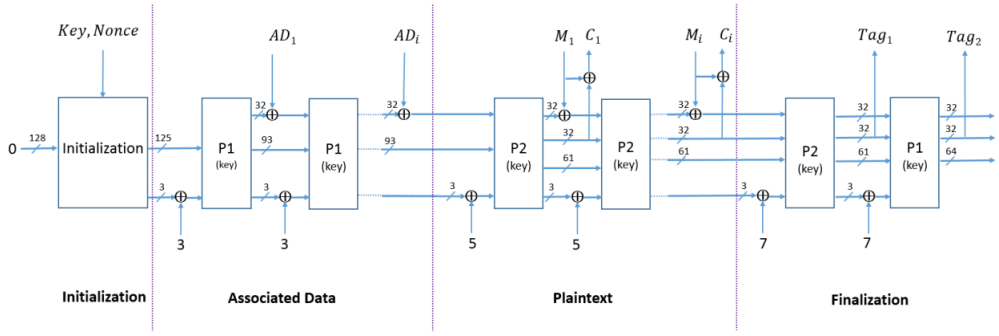

Figure 1: The TinyJAMBU mode for 128-bit state and keyed-permutations [2]

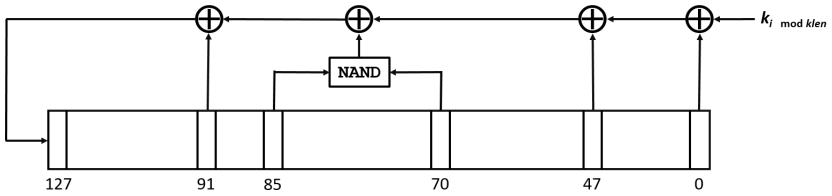

Figure 2: The 128-bit Nonlinear Feedback Shift Register in TinyJAMBU [2]

Nonce setup:

$$\begin{aligned} &\textit{for } i \textit{ from } 0 \textit{ to } 2\textit{:} \\ &\qquad s_{36\ldots38} = s_{36\ldots38} \oplus c_{0\ldots2} \\ &\qquad \textit{Update } \boldsymbol{s} \textit{ with } P_{640}. \\ &\qquad s_{96\ldots127} = s_{96\ldots127} \oplus nonce_{32i\ldots32i+31} \end{aligned} \tag{2}$$

where $c_{0\ldots2}$ are constant bits.

3.2 Proccesing The Associated Data

In this phase, the associated data blocks of 32 bits will be processed and the remaining bits (if exist) will be processed as well.

$$\begin{aligned} &\textit{for } i \textit{ from } 0 \textit{ to } \lfloor adlen/32 \rfloor\textit{:} \\ &\qquad s_{36\ldots38} = s_{36\ldots38} \oplus c_{3\ldots5} \\ &\qquad \textit{Update } \boldsymbol{s} \textit{ with } P_{640}. \\ &\qquad s_{96\ldots127} = s_{96\ldots127} \oplus ad_{32i\ldots32i+31} \\ &\textit{if } (adlen \bmod 32) > 0 : \\ &\qquad s_{36\ldots38} = s_{36\ldots38} \oplus c_{3\ldots5} \\ &\qquad \textit{Update } \boldsymbol{s} \textit{ with } P_{640}. \\ &\qquad s_{96\ldots96+(adlen \bmod 32)} = s_{96\ldots96+(adlen \bmod 32)} \oplus \\ &\qquad\qquad ad_{adlen-(adlen \bmod 32)\ldots adlen-1} \\ &\qquad s_{32\ldots33} = s_{32\ldots33} \oplus ((adlen \bmod 32)/8) \end{aligned} \tag{3}$$

where $c_{3\ldots5}$ are constant bits and $adlen$ is the lenth of associated data in bits.

3.3 The Encryption Phase

In the encryption phase, the message blocks of 32 bits will be processed and the remaining bits (if exist) will be processed as well. The number of rounds r of P_r for key lengths 128, 192, and 256 is 1024, 1152 and 1280, respectively.

$$
\begin{aligned}
&\textit{for } i \textit{ from } 0 \textit{ to } \lfloor mlen/32 \rfloor\textit{:}\\
&\qquad s_{36\ldots38} = s_{36\ldots38} \oplus c_{6\ldots8}\\
&\qquad \textit{Update } \boldsymbol{s} \textit{ with } P_r.\\
&\qquad s_{96\ldots127} = s_{96\ldots127} \oplus m_{32i\ldots32i+31}\\
&\qquad c_{32i\ldots32i+31} = s_{64\ldots95} \oplus m_{32i\ldots32i+31}\\
&\textit{if } (mlen \bmod 32) > 0 :\\
&\qquad s_{36\ldots38} = s_{36\ldots38} \oplus c_{6\ldots8}\\
&\qquad \textit{Update } \boldsymbol{s} \textit{ with } P_r.\\
&\qquad s_{96\ldots96+(mlen \bmod 32)-1} = s_{96\ldots96+(mlen \bmod 32)-1} \oplus\\
&\qquad\qquad m_{mlen-(mlen \bmod 32)\ldots mlen-1}\\
&\qquad c_{mlen-(mlen \bmod 32)\ldots mlen-1} = s_{64\ldots64+(mlen \bmod 32)-1} \oplus\\
&\qquad\qquad m_{mlen-(mlen \bmod 32)\ldots mlen-1}\\
&\qquad s_{32\ldots33} = s_{32\ldots33} \oplus ((mlen \bmod 32)/8)
\end{aligned}
\tag{4}
$$

where $c_{6\ldots8}$ are constant bits and $mlen$ is the length of message in bits.

3.4 The Finalization Phase

In the finalization phase, the 64-bit tag will be generated. The number of rounds r of P_r for key lengths 128, 192, and 256 is 1024, 1152, and 1280, respectively.

$$
\begin{aligned}
&s_{36\ldots38} = s_{36\ldots38} \oplus c_{9\ldots11}\\
&\textit{Update } \boldsymbol{s} \textit{ with } P_r.\\
&t_{0\ldots31} = s_{64\ldots95}\\
&s_{36\ldots38} = s_{36\ldots38} \oplus c_{9\ldots11}\\
&\textit{Update } \boldsymbol{s} \textit{ with } P_{640}.\\
&t_{32\ldots63} = s_{64\ldots95}
\end{aligned}
\tag{5}
$$

where $c_{9\ldots11}$ are constant bits.

In this paper, we focus on the security of the permutation P_n and propose integral distinguishers for different values of n.

4 Integral Distinguisher of TinyJAMBU

Utilizing the automated conventional division property based on MILP models [5], we have studied the security of the permutation P_r in different versions of TinyJAMBU against integral distinguishers. As summarized in Table 3, the results of our studies can be divided into four parts.

- In the first part, we have searched for the maximum number of distinguishable rounds of the permutation P_r with maximum data complexity, i.e. $n-1$ active bits in an n-bit permutation. The results are reported in section 4.1.
- In the second part, we have analyzed the security of P_{1152} which is the permutation with the most number of rounds in different phases of TinyJAMBU-192. The results of this part are reported in section 4.2.

Table 3: Summary of the results

Permutation	Data Compl.	Constant Bits	Balanced Bits	Reference
P_{1273}	2^{127}	s_{127}	s_0	4.1
P_{1152}	2^{127}	$s_i, \forall i \in \{6, 14, 15, ..., 127\}$	s_0	4.2
P_{1024}	2^{127}	$s_i, \forall i \in \{0, ..., 127\}$	s_0	4.3
P_{1024}	2^{126}	$\{s_i, s_{(i+1)}\}, \forall i \in \{5, ..., 126\}$	s_0	4.3
P_{438}	2^{32}	$\{s_0, ..., s_{95}\}$	s_{64}	4.4

- In the third part, we have analyzed the security of P_{1024} which is the permutation with the most number of rounds in different phases of TinyJAMBU-128. The results of this part are reported in section 4.3.
- In the last part, we have searched for the maximum number of distinguishable rounds of the permutation P_r with permissible data according to the TinyJAMBU scheme. In more detail, in this part, we have considered that we only have control over the 32-bit capacity segment of the internal state, i.e. $\{s_{96}, ..., s_{127}\}$ and the bits from the rate segment, i.e. $\{s_{64}, ..., s_{95}\}$, can be monitored in the output. The results of this part are reported in section 4.4.

4.1 Maximum Distinguishable Number of Rounds

We have searched for the maximum number of r where P_r is distinguishable with maximum data complexity, i.e. $n-1$ active bits for an n-bit permutation. The best achieived result for r is 1273. The distinguisher is as follows:

$$\{127\} \xrightarrow{P_{1273}} \{0\}$$

which means that constant bit in s_{127} and active bits in other positions leads to a balanced state in s_0 after applying P_{1273} on $\boldsymbol{s}$.

It deserves to note that P_{1273} consists of 249 and 121 rounds more than the highest number of rounds in P_r in all phases of TinyJAMBU-128 and TinyJAMBU-192, respectively. However, it consists of 7 rounds less than the highest number of rounds in P_r in different phases of TinyJAMBU-256. Moreover, we have searched for integral distinguishers based on conventional division property for P_r for r up to 1287, where 1273 was the highest value for r for which P_r is distinguishable from a random permutation. However, it should be noted that due to the missed detection error in conventional division property, the security of P_r for the studied number of rounds cannot be guaranteed.

4.2 Distingushers on 1152-Round Permutation

We have searched for integral distinguishers for P_{1152}, i.e. permutation with the highest number of rounds in different phases of TinuJAMBU-192, and found 115 distinguishers with data complexity 2^{127}. Distinguishers are as follows:

$$\{i\}, \forall i \in \{6, 14, 15, ..., 127\} \xrightarrow{P_{1152}} \{0\}$$

which means one constant bit in all positions of the initial state, except $\{s_0, s_1, s_2, s_3, s_4, s_5, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}\}$, will lead to balanced property in s_0 after applying P_{1152}.

4.3 Distingushers on 1024-Round Permutation

We have searched for integral distinguishers for P_{1024}, i.e. permutation with the highest number of rounds in different phases of TinuJAMBU-128, and found that with 127 active bits in the input (independent of the constant bit position) the 0^{th} bit in the internal state is balanced after applying P_{1024} on $\boldsymbol{s}$.

$$\{i\}, \forall i \in \{0, ..., 127\} \xrightarrow{P_{1024}} \{0\}$$

Moreover, we have found 122 integral distinguishers for the 0^{th} bit of the internal state after applying P_{1024} with data complexity 2^{126}, where s_i and s_{i+1} for $5 \leq i \leq 126$ are constant and the the others are active in the internal state in the input of P_{1024}.

$$\{i, i+1\}, \forall i \in \{5, ..., 126\} \xrightarrow{P_{1024}} \{0\}$$

4.4 Maximum Distinguishable Number of Rounds with Valid Input Data

We have found that the s_{127} bit in the internal state after applying P_{375} is balanced if $\{s_{96}, s_{97}, ..., s_{127}\}$ (32 bits in the capacity segment) are active and the the others are constant in the internal state in the input of the permutation. This is equivalent to the distinguisher where the bit s_{64} (the last bit in the rate segment) in the internal state after applying P_{438} is balanced.

$$\{0, 1, ..., 95\} \xrightarrow{P_{438}} \{64\}$$

It deserves to note that increasing the round number of permutation P_r from 384 to 640 in the initialization, processing the associated data and finalization phases in the last version has removed the vulnerability against this distinguisher.

5 Conclusion

In this paper, We reported integral distinguisher for different numbers of rounds of the permutation used in the NIST lightweight cryptography competition TinyJAMBU. Distinguishers are found with the conventional division property and despite previous works, do not rely on known-key or related-key setting assumptions although some of our distinguishers are with much more data complexities. It should be noted that according to the designers' justification against previously reported distinguishers, the reported ones in this paper do not treat the security of TinyJAMBU. This is because the distinguishers are either on the reduced versions of the permutation or need impermissible data. However, our target has been analyzing the security of the permutation on its own and we believe that the reported distinguishers can shed some light on the security of TinyJAMBU.

Supporting Data

The codes underlying this article are available in GitHub, at:

https://github.com/khalesiakram/TinyJambuDivision.

References

[1] https://csrc.nist.gov/projects/lightweight-cryptography.

[2] Hongjun Wu and Tao Huang. Tinyjambu: A family of lightweight authenticated encryption algorithms (version 2). *Submission to the NIST Lightweight Cryptography Standardization Process (May 2021)*, 2021.

[3] Yosuke Todo. Structural evaluation by generalized integral property. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 287–314. Springer, 2015.

[4] Yosuke Todo and Masakatu Morii. Bit-based division property and application to simon family. In *International Conference on Fast Software Encryption*, pages 357–377. Springer, 2016.

[5] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying milp method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 648–678. Springer, 2016.

[6] Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier. Cube attacks on non-blackbox polynomials based on division property. In *Advances in Cryptology – CRYPTO 2017*, pages 250–279. Springer International Publishing, 2017.

[7] Kai Hu, Siwei Sun, Yosuke Todo, Meiqin Wang, and Qingju Wang. Massive superpoly recovery with nested monomial predictions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 392–421. Springer, 2021.

[8] Xichao Hu, Yongqiang Li, Lin Jiao, and Mingsheng Wang. New division property propagation table: Applications to block ciphers with large s-boxes. *The Computer Journal*, 2021.

[9] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang. Modeling for three-subset division property without unknown subset. *Journal of Cryptology*, 34(3):1–69, 2021.

[10] Yonglin Hao, Lin Jiao, Chaoyun Li, Willi Meier, Yosuke Todo, and Qingju Wang. Links between division property and other cube attack variants. *IACR Transactions on Symmetric Cryptology*, pages 363–395, 2020.

[11] Yonglin Hao, Takanori Isobe, Lin Jiao, Chaoyun Li, Willi Meier, Yosuke Todo, and Qingju Wang. Improved division property based cube attacks exploiting algebraic properties of superpoly. *IEEE Transactions on Computers*, 68(10), 2019.

[12] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang. Modeling for three-subset division property without unknown subset. volume 34, pages 1–69. Springer, 2021.

[13] Kai Hu and Meiqin Wang. Automatic search for a variant of division property using three subsets. In *Cryptographers' Track at the RSA Conference*, pages 412–432. Springer, 2019.

[14] Kai Hu, Siwei Sun, Meiqin Wang, and Qingju Wang. An algebraic formulation of the division property: Revisiting degree evaluations, cube attacks, and key-independent sums. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 446–476. Springer, 2020.

[15] Zahra Eskandari, Andreas Brasen Kidmose, Stefan Kölbl, and Tyge Tiessen. Finding integral distinguishers with ease. In *International Conference on Selected Areas in Cryptography*, pages 115–138. Springer, 2018.

[16] Qingju Wang, Yonglin Hao, Yosuke Todo, Chaoyun Li, Takanori Isobe, and Willi Meier. Improved division property based cube attacks exploiting algebraic properties of superpoly. In *Annual International Cryptology Conference*, pages 275–305. Springer, 2018.

[17] Senpeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi. Milp-aided method of searching division property using three subsets and applications. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 398–427. Springer, 2019.

[18] Senpeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi. Exploring secret keys in searching integral distinguishers based on division property. *IACR Transactions on Symmetric Cryptology*, pages 288–304, 2020.

[19] Senpeng Wang, Bin Hu, Jie Guan, Kai Zhang, and TaiRong Shi. A practical method to recover exact superpoly in cube attack. *IACR Cryptol. ePrint Arch.*, 2019:259, 2019.

[20] Pranjal Dutta, Mahesh Sreekumar Rajasree, and Santanu Sarkar. Weak-keys and key-recovery attack for tinyjambu. *Scientific Reports*, 12(1):16313, 2022.

[21] Orr Dunkelman, Shibam Ghosh, and Eran Lambooij. Full round zero-sum distinguishers on tinyjambu-128 and tinyjambu-192 keyed-permutation in the known-key setting. In *Progress in Cryptology–INDOCRYPT 2022: 23rd International Conference on Cryptology in India, Kolkata, India, December 11–14, 2022, Proceedings*, pages 349–372. Springer, 2023.

[22] Wil Liam Teng, Iftekhar Salam, Wei-Chuen Yau, Josef Pieprzyk, and Raphaël C-W Phan. Cube attacks on round-reduced tinyjambu. *Scientific Reports*, 12(1):5317, 2022.