Structural Evaluation by Generalized Integral Property*

Yosuke Todo

NTT Secure Platform Laboratories, Tokyo, Japan
todo.yosuke@lab.ntt.co.jp

Abstract. In this paper, we show structural cryptanalyses against two popular networks, i.e., the Feistel Network and the Substitute-Permutation Network (SPN). Our cryptanalyses are distinguishing attacks by an improved integral distinguisher. The integral distinguisher is one of the most powerful attacks against block ciphers, and it is usually constructed by evaluating the propagation characteristic of integral properties, e.g., the ALL or BALANCE property. However, the integral property does not derive useful distinguishers against block ciphers with non-bijective functions and bit-oriented structures. Moreover, since the integral property does not clearly exploit the algebraic degree of block ciphers, it tends not to construct useful distinguishers against block ciphers with low-degree functions. In this paper, we propose a new property called *the division property*, which is the generalization of the integral property. It can effectively construct the integral distinguisher even if the block cipher has non-bijective functions, bit-oriented structures, and low-degree functions. From viewpoints of the attackable number of rounds or chosen plaintexts, the division property can construct better distinguishers than previous methods. Although our attack is a generic attack, it can improve several integral distinguishers against specific cryptographic primitives. For instance, it can reduce the required number of chosen plaintexts for the 10-round distinguisher on Keccak-f from 2^{1025} to 2^{515}. For the Feistel cipher, it theoretically proves that Simon 32, 48, 64, 96, and 128 have 9-, 11-, 11-, 13-, and 13-round integral distinguishers, respectively.

Keywords: Block cipher, Integral distinguisher, Feistel Network, Substitute-Permutation Network, Keccak, Simon, AES-like cipher, Boolean function

1 Introduction

The structural evaluation of cryptographic networks is an important topic of cryptology, and it helps a designer to design strong symmetric key primitives. There are several structural evaluations against the Feistel Network and the Substitute-Permutation Network (SPN) [BS01,IS13,Knu02,LR88,Pat04]. As one direction of the structural evaluation, there are the security evaluation by "the generic attack," which exploits only the feature of the network and does not exploit the particular weaknesses of a specific cipher. It is applicable to large classes of block ciphers, but it is not often effective than the dedicated attack against the specific cipher. This paper focuses on generic attacks against both the Feistel Network and the SPN. The existing generic attack shows that the Feistel Network whose F-functions are chosen from random functions or permutations is vulnerable up to 5 rounds [Pat04,Knu02]. Moreover, Biryukov and Shamir showed that the SPN is vulnerable up to 2.5 rounds [BS01].

Our Contribution This paper shows generic attacks against two networks by improving an integral distinguisher. The integral attack was first proposed by Daemen et al. to

Table 1. The number of required chosen plaintexts to construct r-round integral distinguishers on the SIMON family, Serpent, and KECCAK-f.

Target	$\log_2$(#texts)								Method	Reference
	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	$r=11$	$r=12$	$r=13$		
SIMON 32	17	25	29	31	-	-	-	-	our	Sect. 4.3
	-	-	-	-	-	-	-	-	degree	[Knu94,BC13]
SIMON 48	17	29	39	44	46	47	-	-	our	Sect. 4.3
	17	-	-	-	-	-	-	-	degree	[Knu94,BC13]
SIMON 64	17	33	49	57	61	63	-	-	our	Sect. 4.3
	17	-	-	-	-	-	-	-	degree	[Knu94,BC13]
SIMON 96	17	33	57	77	87	92	94	95	our	Sect. 4.3
	17	33	-	-	-	-	-	-	degree	[Knu94,BC13]
SIMON 128	17	33	65	97	113	121	125	127	our	Sect. 4.3
	17	33	-	-	-	-	-	-	degree	[Knu94,BC13]

Target	$\log_2$(#texts)								Method	Reference
	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$		
Serpent	12	28	84	113	124	-	-	-	our	Sect. 5.3
	28	82	113	123	127	-	-	-	degree	[BCC11]

Target	$\log_2$(#texts)								Method	Reference
	$r=8$	$r=9$	$r=10$	$r=11$	$r=12$	$r=13$	$r=14$	$r=15$		
KECCAK-f	130	258	515	1025	1410	1538	1580	1595	our	Sect. 5.3
	257	513	1025	1409	1537	1579	1593	1598	degree	[BCC11]

evaluate the security of SQUARE [DKR97], and then it was formalized by Knudsen and Wagner [KW02]. Nowadays, many integral distinguishers have been proposed against specific ciphers [KW02,LWZ11,WZ11,YPK02,ZRHD08], and they are often constructed by evaluating the propagation characteristic of integral properties, e.g., the ALL property or the BALANCE property. In this paper, we revisit the integral property, and then introduce *the division property* by generalizing the integral property. The division property can effectively construct integral distinguishers even if block ciphers have non-bijective functions, bit-oriented structures, and low-degree functions.

The Feistel Network is a generic construction to create a (2ℓ)-bit pseudo-random permutation from an ℓ-bit pseudo-random function. We call the ℓ-bit function the F-function, and assume that an attacker can not know the specification of the F-function. Our distinguishing attack can attack up to 3 rounds, and it can attack up to 5 rounds if the F-function is limited to a permutation. Unfortunately, they are not improved compared with the previous ones. However, assuming that the algebraic degree of the F-function is smaller than the bit length of the F-function, our attack can attack more rounds than the previous attacks exploiting the low-degree function. We summarize new integral distinguishers in Appendix B. Although the assumption of our attack is only the algebraic degree of the F-function, it can construct new integral distinguishers on the SIMON family [BSS+13]. Since SIMON has a non-bijective F-function and a bit-oriented structure, it is complicated task to construct the integral distinguisher. The division property theoretically introduces that SIMON 32, 48, 64, 96, and 128 have at least 9-, 11-, 11-, 13-, and 13-round integral distinguishers, respectively. Table 1 shows the comparison between our distinguishers and previous ones.

The SPN consists of an S-Layer and a P-Layer, where the S-Layer has m ℓ-bit bijective S-boxes and the P-Layer has an (ℓm)-bit bijective linear function. The attacker can not know the specifications of the S-boxes and the linear function. Surprisingly, our generic attack becomes able to attack more rounds as the number of S-boxes is larger than the bit

length of the S-box. This fact implies that the design of the P-Layer that can diffuse more outputs of S-boxes may not derive prospective security improvements. We summarize new integral distinguishers in Appendix C. Similar to the result against the Feistel Network, the division property is also useful to construct integral distinguishers against specific cryptographic primitives. For instance, we can reduce the required number of chosen plaintexts for the 7-round distinguisher on Serpent [ABK98] from 2^{127} to 2^{124}. Moreover, for the integral distinguisher on KECCAK-f [DBPA11], we can reduce the required number of chosen plaintexts compared with previous ones constructed by Boura et al. [BCC11]. Table 1 shows the comparison between our distinguishers and previous ones.

Organization This paper is organized as follows: In Sect. 2, we show notations, Boolean functions, and the framework of integral distinguishers. In Sect. 3, we propose the division property by generalizing the integral property, and show the propagation characteristic. In Sect. 4 and Sect. 5, we show new distinguishing attacks on the Feistel Network and the SPN, respectively. In Sect. 6, we show that the division property is also useful to construct the dedicated attack against specific ciphers. As an example, we show new distinguishing attacks on the AES-like cipher. Section 7 concludes this paper.

2 Preliminaries

2.1 Notation

We make the distinction between addition of $\mathbb{F}_2^n$ and addition of $\mathbb{Z}$, and we use $\oplus$ and $+$ as addition of $\mathbb{F}_2^n$ and addition of $\mathbb{Z}$, respectively. For any $a \in \mathbb{F}_2^n$, the i-th element is expressed in $a[i]$ and the hamming weight w_a is calculated as $w_a = \sum_{i=1}^{n} a[i]$. Let $1^n \in \mathbb{F}_2^n$ be a value whose all elements are 1. Moreover, let $0^n \in \mathbb{F}_2^n$ be a value whose all elements are 0.

Subsets $\mathbb{S}_k^n$ and $\mathbb{S}_{\boldsymbol{k}}^{n,m}$ Let $\mathbb{S}_k^n$ be a subset of $\mathbb{F}_2^n$ for any integer $k \in \{0, 1, \ldots, n\}$. The subset $\mathbb{S}_k^n$ is a set of all $a \in \mathbb{F}_2^n$ satisfying $k \leq w_a$, and it is defined as

$$\mathbb{S}_k^n := \{a \in \mathbb{F}_2^n \mid k \leq w_a\}.$$

Let $\mathbb{S}_{\boldsymbol{k}}^{n,m}$ be a subset of $(\mathbb{F}_2^n)^m$ for any vector $\boldsymbol{k} \in (\{0, 1, \ldots, n\})^m$. The subset $\mathbb{S}_{\boldsymbol{k}}^{n,m}$ is a set of all $\boldsymbol{a} \in (\mathbb{F}_2^n)^m$ satisfying $k_i \leq w_{a_i}$, and it is defined as

$$\mathbb{S}_{\boldsymbol{k}}^{n,m} := \{(a_1, a_2, \ldots, a_m) \in (\mathbb{F}_2^n)^m \mid k_i \leq w_{a_i} \text{ for } 1 \leq i \leq m\}.$$

Bit Product Functions π_u and $\pi_{\boldsymbol{u}}$ Let $\pi_u : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function for any $u \in \mathbb{F}_2^n$. Let $x \in \mathbb{F}_2^n$ be an input of π_u, and $\pi_u(x)$ is the AND of $x[i]$ satisfying $u[i] = 1$, namely, it is defined as

$$\pi_u(x) := \prod_{i=1}^{n} x[i]^{u[i]}.$$

Let $\pi_{\boldsymbol{u}} : (\mathbb{F}_2^n)^m \to \mathbb{F}_2$ be a function for any $\boldsymbol{u} \in (\mathbb{F}_2^n)^m$. Let $\boldsymbol{x} \in (\mathbb{F}_2^n)^m$ be an input of $\pi_{\boldsymbol{u}}$, namely, $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ is calculated as

$$\pi_{\boldsymbol{u}}(\boldsymbol{x}) := \prod_{i=1}^{m} \pi_{u_i}(x_i).$$

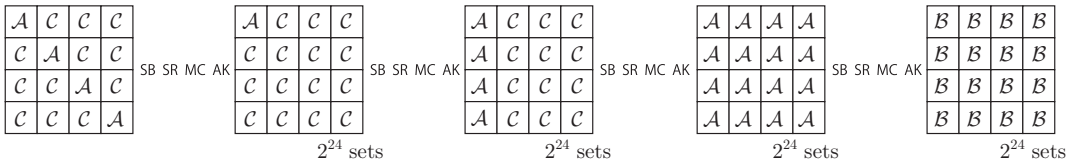

Fig. 1. Integral distinguisher on 4-round AES

2.2 Boolean Function

A Boolean function is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Let $\deg(f)$ be the algebraic degree of a Boolean function f. As representations of the Boolean function, we use Algebraic Normal Form, which is defined as follows.

Algebraic Normal Form Algebraic Normal Form (ANF) is a representation of a Boolean function. Any $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be represented as

$$f(x) = \bigoplus_{u \in \mathbb{F}_2^n} a_u^f \left(\prod_{i=1}^{n} x[i]^{u[i]} \right) = \bigoplus_{u \in \mathbb{F}_2^n} a_u^f \pi_u(x),$$

where $a_u^f \in \mathbb{F}_2$ is a constant value depending on f and u. If $\deg(f)$ is at most d, all a_u^f satisfying $d < w_u$ are 0. An n-bit S-box can be regarded as the concatenation of n Boolean functions. If algebraic degrees of n Boolean functions are at most d, we say the algebraic degree of the S-box is at most d.

2.3 Integral Distinguisher

An integral distinguisher was first proposed by Daemen et al. to evaluate the security of SQUARE [DKR97], and then it was formalized by Knudsen and Wagner [KW02]. It uses a set of chosen plaintexts that contains all possible values for some bits and has a constant value for the other bits. Corresponding ciphertexts are calculated from plaintexts in the set by using an encryption oracle. If the XOR of the corresponding ciphertexts always becomes 0, we say that this cipher has the integral distinguisher.

Integral Property Nowadays, many integral distinguishers have been proposed against specific ciphers [KW02,LWZ11,WZ11,YPK02,ZRHD08], and they are often constructed by evaluating the propagation characteristic of the integral property. We define four integral properties as follows:

- ALL ($\mathcal{A}$) : Every value appears the same number in the multiset.
- BALANCE ($\mathcal{B}$) : The XOR of all texts in the multiset is 0.
- CONSTANT ($\mathcal{C}$) : The value is fixed to a constant for all texts in the multiset.
- UNKNOWN ($\mathcal{U}$) : The multiset is indistinguishable from one of n-bit random values.

Knudsen and Wagner showed that AES has the 4-round integral distinguisher with 2^{32} chosen plaintexts [KW02]. Figure 1 shows the integral distinguisher.

Unfortunately, the integral property does not derive effective distinguishers if block ciphers consist of non-bijective functions, e.g., DES [U.S77] and SIMON [BSS+13] consist

of non-bijection functions. Moreover, since the propagation characteristic does not clearly exploit the algebraic degree of block ciphers, it tends not to construct effective distinguishers against block ciphers with low-degree round functions.

Degree Estimation As another method to construct the integral distinguisher, there is a higher-order differential attack [Lai94,Knu94], which exploits the algebraic degree of block ciphers. When the algebraic degree of a block cipher is at most D, the cipher has the integral distinguisher with 2^{D+1} chosen plaintexts. Canteaut and Videau showed the bound of the degree of iterated round functions [CV02]. Then, Boura et al. improved the bound [BCC11], and showed integral distinguishers on KECCAK [DBPA11] and *Luffa* [CSW08]. We show the bound in Appendix A.

3 Division Property

3.1 Introduction of Division Property

We propose a new property called *the division property*, which is the generalization of the integral property. We consider one bijective S-box with degree d. If an input multiset has $\mathcal{A}$, the output multiset also has $\mathcal{A}$. If an input multiset has $\mathcal{B}$, the output multiset has $\mathcal{U}$. If we have the input multiset with 2^{d+1} chosen texts, the output multiset has $\mathcal{B}$ because the degree of the S-box is d. The integral property does not exploit this property. We now want to exploit useful properties that are hidden between $\mathcal{A}$ and $\mathcal{B}$. Therefore, we redefine $\mathcal{A}$ and $\mathcal{B}$ by the same notation, and then introduce the division property by generalizing the redefinition.

Redefinition of Integral Property Let $\mathbb{X}$ be a multiset whose elements take an n-bit value. We first consider features of the multiset $\mathbb{X}$ satisfying $\mathcal{A}$. If we choose one bit from n bits and calculate the XOR of the chosen bit in the multiset, the calculated value is always 0. Moreover, if we choose at most $(n-1)$ bits from n bits and calculate the XOR of the AND of chosen bits in the multiset, the calculated value is also always 0. However, if we choose all bits from n bits and calculate the XOR of the AND of n bits in the multiset, the calculated value becomes unknown[1]. Above features are expressed by using the bit product function π_u, which is defined in Sect. 2.1, as follows. We evaluate the parity of $\pi_u(x)$ for all $x \in \mathbb{X}$, namely, evaluate $\bigoplus_{x \in \mathbb{X}} \pi_u(x)$. The parity is always even for any u satisfying $w_u < n$. On the other hand, the parity becomes unknown for $u = 1^n$.

We next consider features of the multiset $\mathbb{X}$ satisfying $\mathcal{B}$. If we choose one bit from n bits and calculate the XOR of the chosen bit in the multiset, the calculated value is always 0. However, if we choose at least two bits from n bits and calculate the XOR of the AND of chosen bits in the multiset, the calculated value becomes unknown. Above features are expressed by using the bit product function π_u as follows. We evaluate the parity of $\pi_u(x)$ for all $x \in \mathbb{X}$. The parity is always even for any u satisfying $w_u < 2$. On the other hand, the parity becomes unknown for any u satisfying $w_u \geq 2$.

[1] If all values appear the same even number in the multiset, the calculated value is always 0. If all values appear the same odd number in the multiset, the calculated value is always 1. Thus, we cannot guarantee whether the calculated value is 0 or not when we consider the multiset satisfying $\mathcal{A}$. In this case, we say the calculated value becomes unknown.

3.2 Definition of Division Property

Section 3.1 redefines both the ALL and BALANCE properties by the same notation. Since the redefinition can be parameterized by the number of product bits w_u of the bit product function π_u, we generalize the integral property as follows.

Definition 1 (Division Property). *Let $\mathbb{X}$ be a multiset whose elements take a value of $\mathbb{F}_2^n$, and k takes a value between 0 and n. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}_k^n$, it fulfils the following conditions: The parity of $\pi_u(x)$ for all $x \in \mathbb{X}$ is always even if w_u is less than k. Moreover, the parity becomes unknown if w_u is greater than or equal to k.*

When the multiset $\mathbb{X}$ has $\mathcal{D}_k^n$, it satisfies

$$\bigoplus_{x \in \mathbb{X}} \pi_u(x) = 0, \text{ for all } u \in (\mathbb{F}_2^n \setminus \mathbb{S}_k^n),$$

where $\mathbb{S}_k^n$ is a subset defined in Sect. 2.1. The parity of $\pi_u(x)$ for all $x \in \mathbb{X}$ becomes unknown for any $u \in \mathbb{S}_k^n$. Namely, in the division property, the set of u is divided into the subset that $\bigoplus_{x \in \mathbb{X}} \pi_u(x)$ becomes unknown and the subset that $\bigoplus_{x \in \mathbb{X}} \pi_u(x)$ becomes 0.

Example 1. Let $\mathbb{X}$ be a multiset whose elements take a value of $\mathbb{F}_2^4$. As an example, we prepare the input multiset $\mathbb{X}$ as

$$\mathbb{X} := \{\texttt{0x0}, \texttt{0x3}, \texttt{0x3}, \texttt{0x3}, \texttt{0x5}, \texttt{0x6}, \texttt{0x8}, \texttt{0xB}, \texttt{0xD}, \texttt{0xE}\}.$$

A following table calculates the summation of $\pi_u(x)$.

	0x0	0x3	0x3	0x3	0x5	0x6	0x8	0xB	0xD	0xE	$\sum \pi_u(x)$
	0000	0011	0011	0011	0101	0110	1000	1011	1101	1110	($\bigoplus \pi_u(x)$)
$u = 0000$	1	1	1	1	1	1	1	1	1	1	10 (0)
$u = 0001$	0	1	1	1	1	0	0	1	1	0	6 (0)
$u = 0010$	0	1	1	1	0	1	0	1	0	1	6 (0)
$u = 0011$	0	1	1	1	0	0	0	1	0	0	4 (0)
$u = 0100$	0	0	0	0	1	1	0	0	1	1	4 (0)
$u = 0101$	0	0	0	0	1	0	0	0	1	0	2 (0)
$u = 0110$	0	0	0	0	0	1	0	0	0	1	2 (0)
$u = 0111$	0	0	0	0	0	0	0	0	0	0	0 (0)
$u = 1000$	0	0	0	0	0	0	1	1	1	1	4 (0)
$u = 1001$	0	0	0	0	0	0	0	1	1	0	2 (0)
$u = 1010$	0	0	0	0	0	0	0	1	0	1	2 (0)
$u = 1011$	0	0	0	0	0	0	0	1	0	0	1 (1)
$u = 1100$	0	0	0	0	0	0	0	0	1	1	2 (0)
$u = 1101$	0	0	0	0	0	0	0	0	1	0	1 (1)
$u = 1110$	0	0	0	0	0	0	0	0	0	1	1 (1)
$u = 1111$	0	0	0	0	0	0	0	0	0	0	0 (0)

For all u satisfying $w_u < 3$, $\bigoplus_{x \in \mathbb{X}} \pi_u(x)$ becomes 0. Therefore, the multiset has the division property $\mathcal{D}_3^4$.

Each definition of $\mathcal{B}$ and $\mathcal{U}$ is essentially the same as that of $\mathcal{D}_2^n$ and $\mathcal{D}_1^n$, respectively. However, the definition of $\mathcal{A}$ is different from that of $\mathcal{D}_n^n$. The multiset satisfying $\mathcal{A}$ always has the division property $\mathcal{D}_n^n$ but not vice versa. For instance, the multiset satisfying the

EVEN property, which is defined that the number of occurrences is even for all values [SK12], does not always have $\mathcal{A}$, but it always has $\mathcal{D}_n^n$. In this paper, we use only $\mathcal{D}_n^n$ instead of $\mathcal{A}$ because it is sufficient to use $\mathcal{D}_n^n$ from the viewpoint of the construction of integral distinguishers.

Propagation Characteristic of Division Property Let s be an S-box whose degree is d. Let $\mathbb{X}$ be an input multiset whose elements take a value of $\mathbb{F}_2^n$. Let $\mathbb{Y}$ be an output multiset whose elements are calculated from $s(x)$ for all $x \in \mathbb{X}$. We assume that $\mathbb{X}$ has $\mathcal{D}_k^n$, and want to evaluate the division property of $\mathbb{Y}$. In the division property, the set of u is divided into the subset that $\bigoplus_{x\in\mathbb{X}} \pi_u(x)$ becomes unknown and the subset that $\bigoplus_{x\in\mathbb{X}} \pi_u(x)$ becomes 0. Therefore, we divide the set of v into the subset that $\bigoplus_{s(x)\in\mathbb{Y}} \pi_v(s(x))$ becomes unknown and the subset that $\bigoplus_{s(x)\in\mathbb{Y}} \pi_v(s(x))$ becomes 0. Since the parity of $\pi_v(s(x))$ for all $s(x) \in \mathbb{Y}$ is equal to that of $(\pi_v \circ s)(x)$ for all $x \in \mathbb{X}$, we evaluate $\bigoplus_{x\in\mathbb{X}}(\pi_v \circ s)(x)$.

Proposition 1 (Propagation Characteristic of Division Property). *Let s be an function (S-box) from n bits to n bits, and the degree is d. Assuming that an input multiset $\mathbb{X}$ has the division property $\mathcal{D}_k^n$, the output multiset $\mathbb{Y}$ has $\mathcal{D}_{\lceil \frac{k}{d} \rceil}^n$. In addition, assuming that the S-box is a permutation, the output multiset $\mathbb{Y}$ has $\mathcal{D}_n^n$ when the input multiset has $\mathcal{D}_n^n$.*

Proof. We represent $\bigoplus_{x\in\mathbb{X}}(\pi_v \circ s)(x)$ by using ANF as

$$\begin{aligned}\bigoplus_{x\in\mathbb{X}}(\pi_v \circ s)(x) &= \bigoplus_{x\in\mathbb{X}} \left(\bigoplus_{u\in\mathbb{F}_2^n} a_u^{\pi_v \circ s} \pi_u(x) \right) \\ &= \bigoplus_{u\in\mathbb{S}_k^n} a_u^{\pi_v \circ s} \left(\bigoplus_{x\in\mathbb{X}} \pi_u(x) \right) \oplus \bigoplus_{u\in(\mathbb{F}_2^n \setminus \mathbb{S}_k^n)} a_u^{\pi_v \circ s} \left(\bigoplus_{x\in\mathbb{X}} \pi_u(x) \right).\end{aligned}$$

Since the multiset $\mathbb{X}$ has $\mathcal{D}_k^n$, $\bigoplus_{x\in\mathbb{X}} \pi_u(x)$ is always 0 for any $u \in (\mathbb{F}_2^n \setminus \mathbb{S}_k^n)$. Therefore, it satisfies

$$\bigoplus_{x\in\mathbb{X}}(\pi_v \circ s)(x) = \bigoplus_{u\in\mathbb{S}_k^n} a_u^{\pi_v \circ s} \left(\bigoplus_{x\in\mathbb{X}} \pi_u(x) \right).$$

If $a_u^{\pi_v \circ s}$ is 0 for all $u \in \mathbb{S}_k^n$, $\bigoplus_{x\in\mathbb{A}}(\pi_v \circ s)(x)$ always becomes 0. In other words, if there exists $u \in \mathbb{S}_k^n$ such that $a_u^{\pi_v \circ s}$ is 1, $\bigoplus_{x\in\mathbb{A}}(\pi_v \circ s)(x)$ becomes unknown. Since the function π_v is the AND of w_v bits and the degree of S-box is d, the degree of the Boolean function $(\pi_v \circ s)$ has the following properties:

- The degree of $(\pi_v \circ s)$ is at most $\min\{n, w_v \times d\}$.
- If the S-box is a permutation, the degree of $(\pi_v \circ s)$ is at most $n-1$ for $w_v < n$.

We first assume that the multiset $\mathbb{X}$ has $\mathcal{D}_k^n$. In this case, we consider only u satisfying $w_u \geq k$. When $w_v \times d < k$ holds, $a_u^{\pi_v \circ s}$ is always 0. Thus, the necessary condition that $a_u^{\pi_v \circ s}$ becomes 1 is $w_v \times d \geq k$, and it is $w_v \geq \lceil \frac{k}{d} \rceil$. Namely, the necessary condition that $\bigoplus_{x\in\mathbb{X}}(\pi_v \circ s)(x)$ becomes unknown is $w_v \geq \lceil \frac{k}{d} \rceil$, and $\mathbb{Y}$ has $\mathcal{D}_{\lceil \frac{k}{d} \rceil}^n$. We next assume that the multiset $\mathbb{X}$ has $\mathcal{D}_n^n$ and the S-box is a permutation. In this case, we consider only $u = 1^n$. When $w_v < n$ holds, $a_{1^n}^{\pi_v \circ s}$ is always 0 because the degree of the Boolean function $(\pi_v \circ s)$ is at most $n-1$. Thus, the necessary condition that $a_{1^n}^{\pi_v \circ s}$ becomes 1 is $v = 1^n$. Namely, the necessary condition that $\bigoplus_{x\in\mathbb{X}}(\pi_v \circ s)(x)$ becomes unknown is $v = 1^n$, and $\mathbb{Y}$ has $\mathcal{D}_n^n$. □

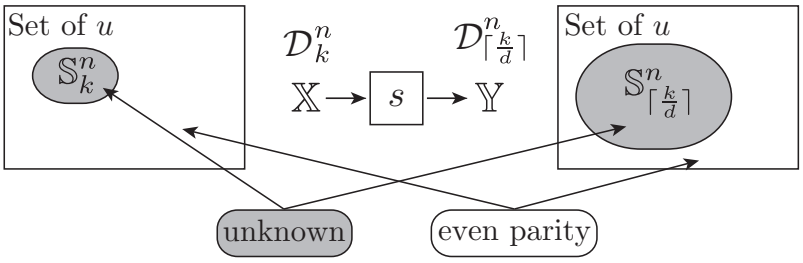

Fig. 2. Propagation characteristic of division property

Example 2. Let us consider a following 4-bit S-box.

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
$s(x)$	0x8	0xC	0x0	0xB	0x9	0xD	0xE	0x5	0xA	0x1	0x2	0x6	0x4	0xF	0x3	0x7

The S-box is bijective and the algebraic degree is 2. We now prepare the input multiset $\mathbb{X}$ as

$$\mathbb{X} := \{\texttt{0x0}, \texttt{0x3}, \texttt{0x3}, \texttt{0x3}, \texttt{0x5}, \texttt{0x6}, \texttt{0x8}, \texttt{0xB}, \texttt{0xD}, \texttt{0xE}\},$$

which is the same as Example 1 and the division property is $\mathcal{D}_3^4$. The output multiset is calculated as

$$\mathbb{Y} := \{\texttt{0x8}, \texttt{0xB}, \texttt{0xB}, \texttt{0xB}, \texttt{0xD}, \texttt{0xE}, \texttt{0xA}, \texttt{0x6}, \texttt{0xF}, \texttt{0x3}\},$$

and a following table calculates the summation of $\pi_v(y)$.

	0x8	0xB	0xB	0xB	0xD	0xE	0xA	0x6	0xF	0x3	$\sum \pi_v(y)$
	1000	1011	1011	1011	1101	1110	1010	0110	1111	0011	$(\bigoplus \pi_v(y))$
$v = 0000$	1	1	1	1	1	1	1	1	1	1	10 (0)
$v = 0001$	0	1	1	1	1	0	0	0	1	1	6 (0)
$v = 0010$	0	1	1	1	0	1	1	1	1	1	8 (0)
$v = 0011$	0	1	1	1	0	0	0	0	1	1	5 (1)
$v = 0100$	0	0	0	0	1	1	0	1	1	0	4 (0)
$v = 0101$	0	0	0	0	1	0	0	0	1	0	2 (0)
$v = 0110$	0	0	0	0	0	1	0	1	1	0	3 (1)
$v = 0111$	0	0	0	0	0	0	0	0	1	0	1 (1)
$v = 1000$	1	1	1	1	1	1	1	0	1	0	8 (0)
$v = 1001$	0	1	1	1	1	0	0	0	1	0	5 (1)
$v = 1010$	0	1	1	1	0	1	1	0	1	0	6 (0)
$v = 1011$	0	1	1	1	0	0	0	0	1	0	4 (0)
$v = 1100$	0	0	0	0	1	1	0	0	1	0	3 (1)
$v = 1101$	0	0	0	0	1	0	0	0	1	0	2 (0)
$v = 1110$	0	0	0	0	0	1	0	0	1	0	2 (0)
$v = 1111$	0	0	0	0	0	0	0	0	1	0	1 (1)

For all v satisfying $w_v < 2$, $\bigoplus_{y \in \mathbb{Y}} \pi_v(y)$ becomes 0. Therefore, the multiset $\mathbb{Y}$ has the division property $\mathcal{D}_2^4$.

Figure 2 shows the outline of the propagation characteristic of the division property. Let $\mathbb{X}$ and $\mathbb{Y}$ be input and output multisets, respectively. First, the size of the set of u that $\bigoplus_{x\in\mathbb{X}} \pi_u(x)$ becomes unknown is small. However, the size of the set of u that $\bigoplus_{x\in\mathbb{X}} \pi_u(s(x))$ becomes unknown expands. If the size expands to the universal set except for 0^n, we regard that the output multiset is indistinguishable from the multiset of random texts.

3.3 Vectorial Division Property

Section 3.2 only shows the division property for one S-box. However, since practical ciphers use several S-boxes in every round, we can not construct integral distinguishers by only using Proposition 1. Therefore, we vectorize the division property.

Let an S-Layer be any function that consists of m n-bit S-boxes with degree d in parallel. We now consider the propagation characteristic of the division property against the S-Layer. Let $\mathbb{X}$ be the input multiset of the S-Layer, and $x \in \mathbb{X}$ takes a value of $(\mathbb{F}_2^n)^m$. The vectorization is the natural extension of the division property. Namely, the set of $\boldsymbol{u}$ is divided into the subset that $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x})$ becomes unknown and the subset that $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x})$ becomes 0, where $\boldsymbol{u}$ is an m-dimensional vector whose elements take a value of $\mathbb{F}_2^n$. Figure 3 shows the difference between the division property and the vectorial one.

Definition 2 (Vectorial Division Property). *Let $\mathbb{X}$ be the multiset whose elements take a value of $(\mathbb{F}_2^n)^m$, and $\boldsymbol{k}$ is an m-dimensional vector whose elements take a value between 0 and n. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}^{n,m}_{\boldsymbol{k}}$, the multiset fulfils the following conditions: The parity of $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{X}$ is always even if $\boldsymbol{u}$ does NOT belong to $\mathbb{S}^{n,m}_{\boldsymbol{k}}$. Moreover, the parity becomes unknown if $\boldsymbol{u}$ belongs to $\mathbb{S}^{n,m}_{\boldsymbol{k}}$.*

Propagation Characteristic of Vectorial Division Property Assume that the input multiset of the S-Layer has the division property $\mathcal{D}^{n,m}_{\boldsymbol{k}}$. The output of the S-Layer is calculated as $S(\boldsymbol{x}) = (s_1(x_1), s_2(x_2), \ldots, s_m(x_m))$ for $(x_1, x_2, \ldots, x_m) \in \mathbb{X}$. We now consider the set of $\boldsymbol{v}$ that $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{v}}(S(\boldsymbol{x}))$ becomes unknown and the set of $\boldsymbol{v}$ that $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{v}}(S(\boldsymbol{x}))$ becomes 0. Since the output of each S-box is calculated independently, the propagation characteristic of the division property can also be evaluated independently. Namely, the output multiset has $\mathcal{D}^{n,m}_{\boldsymbol{k}'}$, where $k'_i = \lceil k_i/d \rceil$ holds. Moreover, if the S-box is bijective and $k_i = n$ holds, $k'_i = n$ holds.

3.4 Collective Division Property

By vectorizing of the division property, we can evaluate the multiset whose elements take a value of $(\mathbb{F}_2^n)^m$. However, it is still insufficient to use only vectorial division property. For simplicity, we consider a multiset $\mathbb{X}$ whose elements take a value of $(\mathbb{F}_2^8)^2$. Assume that the number of elements in $\mathbb{X}$ is 256, and two elements of $\boldsymbol{x}$ take all values from 0 to 255 independently. We consider the set of $\boldsymbol{u}$ that the parity of $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{X}$ becomes unknown and the set of $\boldsymbol{u}$ that the parity becomes 0.

- The parity becomes unknown if $\boldsymbol{u}$ belongs to $\mathbb{S}^{8,2}_{[8,0]}$.
- The parity becomes unknown if $\boldsymbol{u}$ belongs to $\mathbb{S}^{8,2}_{[0,8]}$.
- The parity becomes unknown if $\boldsymbol{u}$ belongs to $\mathbb{S}^{8,2}_{[1,1]}$.
- Otherwise, i.e., $\boldsymbol{u}$ does NOT belong to $\mathbb{S}^{8,2}_{[8,0]} \cup \mathbb{S}^{8,2}_{[0,8]} \cup \mathbb{S}^{8,2}_{[1,1]}$, the parity is always even.

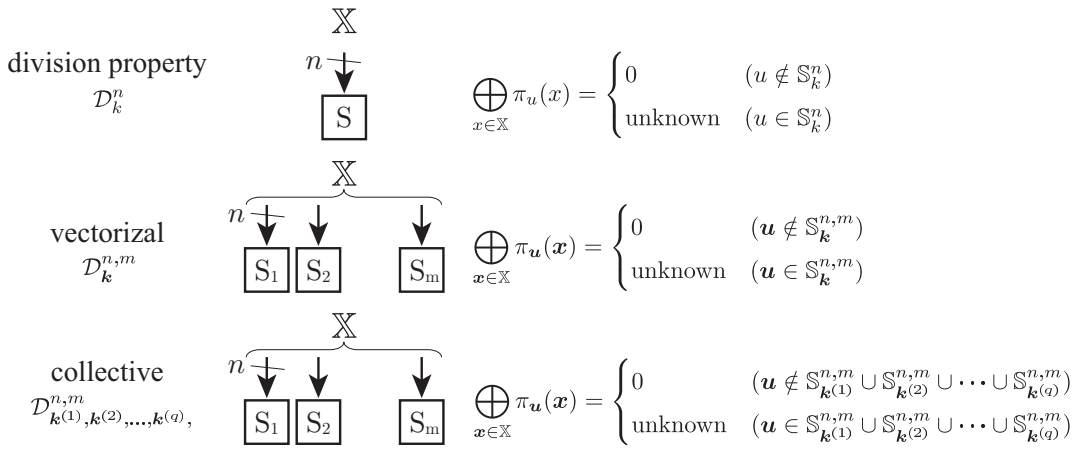

Fig. 3. Division property, vectorial division property, and collective division property

We can not express this property by using the vectorial division property. Therefore, we collect several vectorial division properties. Figure 3 shows the difference between the vectorial division property and the collective division property.

Definition 3 (Collective Division Property). *Let $\mathbb{X}$ be the multiset whose elements take a value of $(\mathbb{F}_2^n)^m$, and $\boldsymbol{k}^{(j)}$ $(j = 1, 2, \ldots, q)$ are m-dimensional vectors whose elements take a value between 0 and n. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}^{n,m}_{\boldsymbol{k}^{(1)},\boldsymbol{k}^{(2)},\ldots,\boldsymbol{k}^{(q)}}$, the multiset fulfils the following conditions: The parity of $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{X}$ is always even if $\boldsymbol{u}$ does NOT belong to the union $\mathbb{S}^{n,m}_{\boldsymbol{k}^{(1)}} \cup \mathbb{S}^{n,m}_{\boldsymbol{k}^{(2)}} \cup \cdots \cup \mathbb{S}^{n,m}_{\boldsymbol{k}^{(q)}}$. Moreover, the parity becomes unknown if $\boldsymbol{u}$ belongs to the union $\mathbb{S}^{n,m}_{\boldsymbol{k}^{(1)}} \cup \mathbb{S}^{n,m}_{\boldsymbol{k}^{(2)}} \cup \cdots \cup \mathbb{S}^{n,m}_{\boldsymbol{k}^{(q)}}$.*

It is obvious that the collective division property with $q = 1$ is the same as the vectorial division property.

Propagation Characteristic of Collective Division Property Assume that the input multiset of the S-Layer has the division property $\mathcal{D}^{n,m}_{\boldsymbol{k}^{(1)},\boldsymbol{k}^{(2)},\ldots,\boldsymbol{k}^{(q)}}$. We now consider the set of $\boldsymbol{v}$ that $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{v}}(S(\boldsymbol{x}))$ becomes unknown, and the set is derived from only the set of $\boldsymbol{u}$ that $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x})$ becomes unknown. Therefore, we can evaluate the propagation characteristic of $\mathcal{D}^{n,m}_{\boldsymbol{k}^{(j)}}$ for all j independently. Namely, the output multiset has $\mathcal{D}^{n,m}_{\boldsymbol{k}'^{(1)},\boldsymbol{k}'^{(2)},\ldots,\boldsymbol{k}'^{(q)}}$, where $k_i'^{(j)} = \lceil k_i^{(j)}/d \rceil$ holds. Moreover, if the S-box is bijective and $k_i^{(j)} = n$ holds, $k_i'^{(j)} = n$ holds.

4 Improved Integral Distinguishers on Feistel Network

4.1 Feistel Network

(ℓ, d)-Feistel The Feistel Network is one of the most popular network to design block ciphers. When n-bit block ciphers are constructed by the Feistel Network, the input of the round function is expressed in two $(n/2)$-bit values. Moreover, an $(n/2)$-bit non-linear function F is used in the round function, and we call this function the F-function. Let (w_1, w_2) be the input of the round function, and the output is calculated as $(z_1, z_2) = (F(w_1) \oplus w_2, w_1)$. We now define an (ℓ, d)-Feistel, whose F-function is an ℓ-bit non-linear function with degree d (this function is not limited to a permutation). Figure 4 shows the

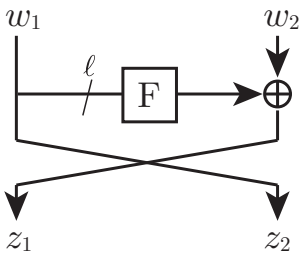

Fig. 4. (ℓ, d)-Feistel

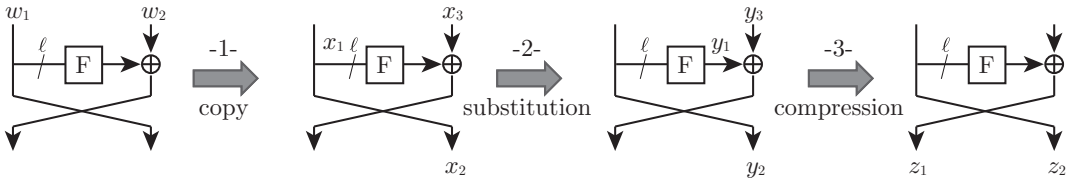

Fig. 5. Propagation characteristic for Feistel Network

round function of the Feistel Network. There are many block ciphers adopting (ℓ, d)-Feistel, e.g. DES [U.S77], Camellia [AIK+00], and SIMON $2n$ [BSS+13] adopt $(32, 5)$-, $(64, 7)$-, and $(n, 2)$-Feistel, respectively.

4.2 Propagation Characteristic for Feistel Network

This section shows that the division property is useful to construct integral distinguishers on (ℓ, d)-Feistel. Since the Feistel Network has "copy," "substitution," and "compression," we need to propagate the division property against them. The "copy" creates the input of the F-function, and the "substitution" processes the input by the F-function, and finally the "compression" creates the left half of the output by XOR. Figure 5 shows the outline of the propagation characteristic.

-1- Copy Let $\mathbb{W}$ be an input set, and $(w_1, w_2) \in \mathbb{W}$ denotes the input value. The round function first creates (x_1, x_2, x_3), where $x_1 = w_1$, $x_2 = w_1$, and $x_3 = w_2$ hold. Here, x_1 is the input of the F-function, x_2 is the right half of the output of the round function, and x_3 is the right half of the input of the round function. Let $\mathbb{X}$ be the output set whose elements take (x_1, x_2, x_3) for all $(w_1, w_2) \in \mathbb{W}$. Assume that the input set $\mathbb{W}$ has the division property $\mathcal{D}^{\ell,2}_{\boldsymbol{k}^{(1)},\boldsymbol{k}^{(2)},\ldots,\boldsymbol{k}^{(q)}}$. If we use $\pi_{\boldsymbol{u}}$ satisfying $k_1^{(j)} \leq u_1$ and $k_2^{(j)} \leq u_2$, the parity of $\pi_{\boldsymbol{u}}(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathbb{W}$ becomes unknown. Since x_1 is equal to x_2, the parity of $\pi_{\boldsymbol{v}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{X}$ becomes unknown if we use $\pi_{\boldsymbol{v}}$ satisfying $k_1^{(j)} - k' \leq v_1$, $k' \leq v_2$, and $k_2^{(j)} \leq v_3$ for all k' $(0 \leq k' \leq k_1^{(j)})$. Therefore, the set $\mathbb{X}$ has the division property $\mathcal{D}^{\ell,3}_{[0,k_1^{(1)},k_2^{(1)}],[1,k_1^{(1)}-1,k_2^{(1)}],\ldots,[k_1^{(1)},0,k_2^{(1)}],\ldots,[k_1^{(q)},0,k_2^{(q)}]}$.

-2- Substitution The F-function is an ℓ-bit function with degree d. Assume that the input set has the division property $\mathcal{D}^{\ell,3}_{\boldsymbol{k}^{(1)},\boldsymbol{k}^{(2)},\ldots,\boldsymbol{k}^{(q)}}$. From the propagation characteristic of the division property, the output set has $\mathcal{D}^{\ell,3}_{\boldsymbol{k}'^{(1)},\boldsymbol{k}'^{(2)},\ldots,\boldsymbol{k}'^{(q)}}$, where $(k_1'^{(j)}, k_2'^{(j)}, k_3'^{(j)}) =$

$(\lceil k_1^{(j)}/d \rceil, k_2^{(j)}, k_3^{(j)})$ holds. If the F-function is limited to a permutation, $k_1^{\prime(j)}$ becomes ℓ when $k_1^{(j)} = \ell$ holds.

-3- Compression Let $\mathbb{Y}$ be the input set, and $(y_1, y_2, y_3) \in \mathbb{Y}$ denotes the input value, where y_1 denotes the output of the F-function. Let y_1 be XORed with y_3, and then the internal state is expressed in $(z_1, z_2) = (y_1 \oplus y_3, y_2)$. Let $\mathbb{Z}$ be the set whose elements take (z_1, z_2) for all $(y_1, y_2, y_3) \in \mathbb{Y}$. To evaluate the division property of the set $\mathbb{Z}$, we calculate the parity of $\pi_{\boldsymbol{v}}(z_1, z_2)$ for all $(z_1, z_2) \in \mathbb{Z}$ as

$$\begin{aligned}
\bigoplus_{(z_1,z_2)\in\mathbb{Z}} \pi_{\boldsymbol{v}}(z_1, z_2) &= \bigoplus_{(z_1,z_2)\in\mathbb{Z}} (\pi_{v_1}(z_1) \times \pi_{v_2}(z_2)) \\
&= \bigoplus_{(y_1,y_2,y_3)\in\mathbb{Y}} (\pi_{v_1}(y_1 \oplus y_3) \times \pi_{v_2}(y_2)) \\
&= \bigoplus_{(y_1,y_2,y_3)\in\mathbb{Y}} \left(\bigoplus_{c \preceq v_1} (\pi_c(y_1) \times \pi_{v_1 \oplus c}(y_3)) \times \pi_{v_2}(y_2) \right) \\
&= \bigoplus_{c \preceq v_1} \left(\bigoplus_{(y_1,y_2,y_3)\in\mathbb{Y}} \pi_c(y_1) \times \pi_{v_2}(y_2) \times \pi_{v_1 \oplus c}(y_3) \right),
\end{aligned}$$

where the set of c chosen from $c \preceq v_1$ denotes the set of c satisfying $c \wedge v_1 = c$. Assuming that the input set $\mathbb{Y}$ has the division property $\mathcal{D}^{\ell,3}_{\boldsymbol{k}^{(1)},\boldsymbol{k}^{(2)},\ldots,\boldsymbol{k}^{(q)}}$, the output set $\mathbb{Z}$ has the division property $\mathcal{D}^{\ell,2}_{\boldsymbol{k}'^{(1)},\boldsymbol{k}'^{(2)},\ldots,\boldsymbol{k}'^{(q)}}$, where $(k_1^{\prime(j)}, k_2^{\prime(j)}) = (k_1^{(j)} + k_3^{(j)}, k_2^{(j)})$ holds. Notice that the parity of $\pi_{\boldsymbol{v}}(z_1, z_2)$ for all $(z_1, z_2) \in \mathbb{Z}$ becomes 0 if $k_1^{(j)} + k_3^{(j)}$ is more than ℓ.

4.3 Path Search Algorithm for (ℓ, d)-Feistel

This section shows the path search algorithm for integral distinguishers against (ℓ, d)-Feistel. The algorithm is based on the propagation characteristic shown in Sect. 4.2. Assume that k_1 bits of the left half of the input are active and the rest $(\ell - k_1)$ bits are constant. Moreover, assume that k_2 bits of the right half of the input are active and the rest $(\ell - k_2)$ bits are constant. Namely, we prepare $2^{k_1+k_2}$ chosen plaintexts. The input set has the division property $\mathcal{D}^{\ell,2}_{[k_1,k_2]}$. Algorithm 1 shows the path search algorithm to create the integral distinguisher on (ℓ, d)-Feistel. Algorithm 1 does not limit the F-function to be a permutation. If the F-function is limited to be a permutation, L becomes $k_2 + \ell$ when $X = \ell$ holds (see the 4-th line in Algorithm 1). Algorithm 1 calls `SizeReduce`, which eliminates $\boldsymbol{k}^{(i,j)}$ if there exists (i', j') satisfying $\mathbb{S}^{\ell,2}_{\boldsymbol{k}^{(i,j)}} \subseteq \mathbb{S}^{\ell,2}_{\boldsymbol{k}^{(i',j')}}$.

Results Table 2 shows the number of required chosen plaintexts to construct r-round integral distinguishers on $(32, 5)$- and $(64, 7)$-Feistel, where DES [U.S77] is classified into $(32, 5)$-Feistel with non-bijective function and Camellia [AIK+00] is classified into $(64, 7)$-Feistel with bijective function. When we construct the integral distinguisher on (ℓ, d)-Feistel with 2^D chosen plaintexts, we use (k_1, k_2) satisfying

$$(k_1, k_2) = \begin{cases} (D - \ell, \ell) & \text{for } \ell \leq D, \\ (0, D) & \text{for } D < \ell. \end{cases}$$

Algorithm 1 Path search algorithm for integral distinguishers on (ℓ, d)-Feistel

procedure `FeistelFuncEval`(ℓ, d, k_1, k_2)
  $q \Leftarrow 0$
  for $X = 0$ to k_1 **do**
    $L \Leftarrow k_2 + \lceil X/d \rceil$
    if $L \leq \ell$ **then**
      $q \Leftarrow q + 1$
      $\boldsymbol{k}^{(q)} \Leftarrow (L, k_1 - X)$
    end if
  end for
  return $\boldsymbol{k}^{(1)}, \ldots, \boldsymbol{k}^{(q)}$
end procedure

procedure `IntegralPathSearch`$(\ell, d, r = 0, k_1, k_2)$
  $\boldsymbol{k}^{(1)}, \ldots, \boldsymbol{k}^{(q)} \Leftarrow$ `FeistelFuncEval`(ℓ, d, k_1, k_2)
  $D \Leftarrow \max\{k_1^{(1)} + k_2^{(1)}, k_1^{(2)} + k_2^{(2)}, \ldots, k_1^{(q)} + k_2^{(q)}\}$
  while $1 < D$ **do**
    $r \Leftarrow r + 1$
    for $i = 1$ to q **do**
      $\boldsymbol{k}^{(i,1)}, \ldots, \boldsymbol{k}^{(i,p_i)} \Leftarrow$ `FeistelFuncEval`$(\ell, d, k_1^{(i)}, k_2^{(i)})$
    end for
    $(\boldsymbol{k}^{(1)}, \boldsymbol{k}^{(2)}, \ldots, \boldsymbol{k}^{(q')}) \Leftarrow$ `SizeReduce`$(\boldsymbol{k}^{(1,1)}, \boldsymbol{k}^{(1,2)}, \ldots, \boldsymbol{k}^{(q,p_q)})$
    $D \Leftarrow \max\{k_1^{(1)} + k_2^{(1)}, k_1^{(2)} + k_2^{(2)}, \ldots, k_1^{(q')} + k_2^{(q')}\}$
    $q \Leftarrow q'$
  end while
  return r
end procedure

For the comparison with our integral distinguishers, we consider two previous methods, one is the propagation characteristic of the integral property and another is the estimation of the algebraic degree. We first consider the propagation characteristic of the integral property. If the F-function is a non-bijective function, the propagation characteristic does not construct sufficient distinguishers. Therefore, results introduced by the integral property are only shown when the F-function is bijective. We next consider the estimation of the algebraic degree. Unfortunately, since we do not know the improved bound against the Feistel Network, we use the trivial bound for the Feistel Network. Assume that the left half of the plaintext is constant. For any r-round (ℓ, d)-Feistel, it can be observed that the function, which associates the right half of the ciphertext with the right half of the plaintext, has degree at most d^{r-2} for $2 \leq r$. Therefore, we can construct the r-round integral distinguishers with $2^{d^{r-2}+1}$ chosen plaintexts. Since the right half of the plaintext is at most ℓ bits, the distinguisher can be constructed with $2^{d^{r-2}+1} < 2^{\ell}$.

As a result, as far as we try, all distinguishers constructed by the division property are "better" than those by previous methods. We summarize integral distinguishers on other (ℓ, d)-Feistel in Appendix B. We already know a "better" integral distinguisher on Camellia in [YPK02], but it is constructed by using the specific feature of Camellia. On the other hand, our method is generic distinguishing attacks against (ℓ, d)-Feistel. From the result of $(64, 7)$-Feistel, it shows that even if the F-function of Camellia is chosen from any functions with degree 7, the modified Camellia has the 6-round integral distinguisher.

Integral Distinguishers on Simon Family Although our attack is a generic attack, it can create new integral distinguishers on the SIMON family [BSS+13]. SIMON is a lightweight

Table 2. The number of chosen plaintexts to construct r-round integral distinguishers on $(32, 5)$- and $(64, 7)$-Feistel. Our distinguishers are got by implementing Algorithm 1.

Target [Application]	F-function	$\log_2$(#texts)						Method	Reference
		$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$		
$(32, 5)$-Feistel [DES]	non-bijection	26	51	62	-	-	-	our	Sect. 4.3
		26	-	-	-	-	-	degree	[Knu94,BC13]
$(64, 7)$-Feistel [Camellia]	bijection	50	98	124	-	-	-	our	Sect. 4.3
		50	-	-	-	-	-	degree	[Knu94,BC13]
		64	-	-	-	-	-	integral	[KW02]

Table 3. The number of chosen plaintexts to construct r-round integral distinguishers on the SIMON family, where the F-function is not bijective. Our distinguishers are got by implementing Algorithm 1.

Target [Application]	$\log_2$(#texts)								Method	Reference
	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	$r=11$	$r=12$	$r=13$		
$(16, 2)$-Feistel [SIMON 32]	17	25	29	31	-	-	-	-	our	Sect. 4.3
	-	-	-	-	-	-	-	-	degree	[Knu94,BC13]
$(24, 2)$-Feistel [SIMON 48]	17	29	39	44	46	47	-	-	our	Sect. 4.3
	17	-	-	-	-	-	-	-	degree	[Knu94,BC13]
$(32, 2)$-Feistel [SIMON 64]	17	33	49	57	61	63	-	-	our	Sect. 4.3
	17	-	-	-	-	-	-	-	degree	[Knu94,BC13]
$(48, 2)$-Feistel [SIMON 96]	17	33	57	77	87	92	94	95	our	Sect. 4.3
	17	33	-	-	-	-	-	-	degree	[Knu94,BC13]
$(64, 2)$-Feistel [SIMON 128]	17	33	65	97	113	121	125	127	our	Sect. 4.3
	17	33	-	-	-	-	-	-	degree	[Knu94,BC13]

block ciphers proposed by the National Security Agency. Since SIMON has a non-bijective F-function and a bit-oriented structure, it is complicated task to construct the integral distinguisher. The division property theoretically shows that SIMON 32, 48, 64, 96, and 128 have at least 9-, 11-, 11-, 13-, and 13-round integral distinguishers, respectively. Table 3 shows the comparison between our distinguishers and previous ones by the degree estimation. On the other hand, Wang et al. showed that SIMON 32 has the 15-round integral distinguisher by experiments [WLV+14]. Therefore, there are 6-round differences between our theoretical result and Wang's experimental result. Our distinguisher is valid against all $(32, 2)$-Feistel and it does not exploit the feature of the round function. Namely, we expect that the 6-round difference is derived from the specification of the round function of SIMON 32.

5 Improved Integral Distinguishers on Substitute-Permutation Network

5.1 Substitute-Permutation Network

(ℓ, d, m)-SPN The Substitute-Permutation Network (SPN) is another important structure for block ciphers. The SPN has a round function that consists of an S-Layer and a P-Layer, and a block cipher is designed by iterating the round function. We now define an (ℓ, d, m)-SPN, whose round function has m ℓ-bit S-boxes in the S-Layer and one (ℓm)-bit linear function in the P-Layer. Here, each S-box is any bijective function whose degree is at most d, and an (ℓm)-bit linear function is any bijective function whose degree is at most 1. Figure 6 shows the round function of the SPN. Nowadays, many block ciphers adopting (ℓ, d, m)-SPN have been proposed, e.g. AES [U.S01], PRESENT [BKL+07],

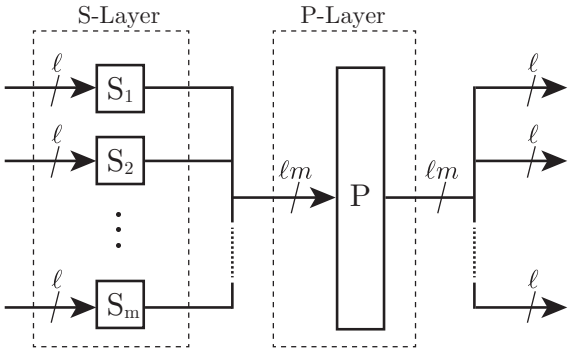

Fig. 6. (ℓ, d, m)-SPN

and Serpent [ABK98] adopt $(8, 7, 16)$-, $(4, 3, 16)$-, and $(4, 3, 32)$-SPN, respectively. Moreover, KECCAK-f [DBPA11], which is a permutation in the hash function KECCAK, can be regarded as $(5, 2, 320)$-SPN.

5.2 Propagation Characteristic for SPN

This section shows that the division property is useful to construct integral distinguishers on (ℓ, d, m)-SPN. We first prepare the set of the input of the S-Layer such that k_i bits of the input of the i-th S-box are active and the rest $(\ell - k_i)$ bits are constant. In this case, the input set has the division property $\mathcal{D}^{\ell,m}_{\boldsymbol{k}}$. We first evaluate the propagation characteristic against the S-Layer. Next, the P-Layer is applied but the input and output take a value of $\mathbb{F}_2^{\ell m}$. Therefore, we need to convert the division property $\mathcal{D}^{\ell,m}_{\boldsymbol{k}}$ into $\mathcal{D}^{\ell m}_{k}$, and then evaluate the propagation characteristic against the P-Layer. Since the S-Layer is applied again after the P-Layer, we convert the division property $\mathcal{D}^{\ell m}_{k}$ into $\mathcal{D}^{\ell,m}_{\boldsymbol{k}^{(1)},\boldsymbol{k}^{(2)},\ldots,\boldsymbol{k}^{(q)}}$. After the second round, we evaluate the propagation characteristic of this collective division property.

- **S-Layer** Assume that the input set of the S-Layer has the division property $\mathcal{D}^{\ell,m}_{\boldsymbol{k}}$. Since the S-Layer consists of m ℓ-bit S-boxes with degree d, the output set of the S-Layer has $\mathcal{D}^{\ell,m}_{\boldsymbol{k}'}$. Here, if $k_i < \ell$ holds, k'_i is calculated as $k'_i = \lceil k_i/d \rceil$. If $k_i = \ell$ holds, k'_i is calculated as $k'_i = \ell$.
- **Concatenation (Conversion form S-Layer to P-Layer)** The output of the S-Layer is expressed in a value of $(\mathbb{F}_2^{\ell})^m$, but the input of the P-Layer is expressed in a value of $\mathbb{F}_2^{\ell m}$. Let $\mathbb{X}$ be the output set of the S-Layer whose elements take a value of $(\mathbb{F}_2^{\ell})^m$. Let $\mathbb{Y}$ be the input set of the P-Layer whose elements take a value of $\mathbb{F}_2^{\ell m}$. The transformation is generally implemented by a simple bit concatenation, namely, $y = (x_1 \| x_2 \| \cdots \| x_m)$ where $(x_1, x_2, \ldots, x_m)$ and y are values of $\mathbb{X}$ and $\mathbb{Y}$, respectively. We now consider the conversion of the division property from $\mathcal{D}^{\ell,m}_{\boldsymbol{k}}$ to $\mathcal{D}^{\ell m}_{k'}$. The parity of $\pi_v(y)$ for all $y \in \mathbb{Y}$ becomes unknown if and only if we choose v satisfying $w_v \geq \sum_{i=1}^{m} k_i$. Therefore, the input set of the P-Layer has the division property $\mathcal{D}^{\ell m}_{k'}$, where $k' = \sum_{i=1}^{m} k_i$ holds.
- **P-Layer** The P-Layer consists of an (ℓm)-bit linear function. Since the degree of the linear function is at most 1, there is no change in the division property.
- **Partition (Conversion form P-Layer to S-Layer)** The output of the P-Layer is expressed in a value of $\mathbb{F}_2^{\ell m}$, but the input of the S-Layer is expressed in a value of $(\mathbb{F}_2^{\ell})^m$. Let $\mathbb{X}$ be the output set of the P-Layer whose elements take a value of $\mathbb{F}_2^{\ell m}$. Let $\mathbb{Y}$ be the input set of the S-Layer whose elements take a value of $(\mathbb{F}_2^{\ell})^m$. The transformation is

Algorithm 2 Path search algorithm for integral distinguishers on (ℓ, d, m)-SPN

```
1: procedure IntegralPathSearch(ℓ, d, m, r = 0, k_1, k_2, ..., k_m)
2:     if k_i < ℓ then k_i ⇐ ⌈k_i/d⌉                          ▷ 1-st round S-Layer
3:     end if
4:     k ⇐ Σ_{i=1}^{m} k_i                                    ▷ 1-st round Concatenation and P-Layer
5:     while 1 < k do
6:         r ⇐ r + 1
7:         if k ≤ (ℓ − 1)m then k ⇐ ⌈k/d⌉                      ▷ (r + 1)-th round
8:         else k ⇐ ⌈(ℓ−1)/d⌉ (ℓm − k) + ℓ(m − ℓm + k)          ▷ (r + 1)-th round
9:         end if
10:    end while
11:    return r
12: end procedure
```

generally implemented by a simple bit partition, namely, $(y_1 \| y_2 \| \cdots \| y_m) = x$ where x and $(y_1, y_2, \ldots, y_m)$ are values of $\mathbb{X}$ and $\mathbb{Y}$, respectively. We now consider the conversion of the division property from $\mathcal{D}_k^{\ell m}$ to $\mathcal{D}_{\boldsymbol{k}'}^{\ell,m}$. When the output set of the P-Layer has $\mathcal{D}_k^{\ell m}$, the sufficient condition that the parity of $\pi_u(x)$ for all $x \in \mathbb{X}$ becomes unknown is $k \leq w_u$. Therefore, the input set of the S-Layer has the collective division property $\mathcal{D}^{\ell,m}_{\boldsymbol{k}'^{(1)},\boldsymbol{k}'^{(2)},\ldots,\boldsymbol{k}'^{(q)}}$, where q denotes the number of all possible vectors satisfying $k_1'^{(j)} + k_2'^{(j)} + \cdots + k_m'^{(j)} = k$ $(1 \leq j \leq q)$. After the second round, we evaluate the propagation characteristic of the collective division property.

We can construct the integral distinguisher by evaluating the propagation characteristic of the collective division property. However, since the size of q extremely expands, it is infeasible to execute the straightforward implementation. Therefore, we show more efficient technique. Let $\mathbb{X}$ be the input set of the S-Layer, and the elements take a value of $(\mathbb{F}_2^\ell)^m$. Assume that the input set has the division property $\mathcal{D}^{\ell,m}_{\boldsymbol{k}^{(1)},\boldsymbol{k}^{(2)},\ldots,\boldsymbol{k}^{(q)}}$ that is created by the partition of the division property $\mathcal{D}_k^{\ell m}$. If $k > (\ell-1)m$ holds, at least $(m-\ell m+k)$ elements of $\boldsymbol{k}^{(j)}$ have to become ℓ. In this case, the rest elements have to become $\ell - 1$. Since the S-Layer derives $\left\lceil \frac{\ell-1}{d} \right\rceil$ and ℓ from $(\ell-1)$ and ℓ, respectively, the output set has the division property $\mathcal{D}_{k'}^{\ell m}$, where k' is calculated as

$$k' = \begin{cases} \left\lceil \frac{\ell-1}{d} \right\rceil (\ell m - k) + \ell(m - \ell m + k) & \text{for } k > (\ell-1)m, \\ \left\lceil \frac{k}{d} \right\rceil & \text{for } k \leq (\ell-1)m. \end{cases}$$

Here, if $k \leq (\ell-1)m$ holds, we simply regard the round function of (ℓ, d, m)-SPN as one (ℓm)-bit S-box with degree d.

5.3 Path Search Algorithm for (ℓ, d, m)-SPN

We now consider integral distinguishers on (ℓ, d, m)-SPN. We first prepare the set of chosen plaintexts such that k_i bits of the input of the i-th S-box are active and the rest $(\ell - k_i)$ bits are constant. Namely, we prepare $2^{\sum_{i=1}^m k_i}$ chosen plaintexts. The input set has the division property $\mathcal{D}_{\boldsymbol{k}}^{\ell,m}$. Algorithm 2 shows the path search algorithm to construct the integral distinguisher.

Results Table 4 shows the number of required chosen plaintexts to construct the r-round integral distinguisher on $(4, 3, 16)$- and $(8, 7, 16)$-SPN, where PRESENT [BKL+07] and

Table 4. The number of chosen plaintexts to construct r-round integral distinguishers on (ℓ, d, m)-SPN. Our distinguishers are got by implementing Algorithm 2.

Target	$\log_2$(#texts)					Method	Reference
	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$		
$(4,3,16)$-SPN	12	28	52	60	-	our	Sect. 5.3
[PRESENT]	28	52	60	63	-	degree	[BCC11]
$(8,7,16)$-SPN	56	120	-	-	-	our	Sect. 5.3
[AES]	117	127	-	-	-	degree	[BCC11]

Table 5. The number of chosen plaintexts to construct r-round integral distinguishers on KECCAK-f and Serpent. Our distinguishers are got by implementing Algorithm 2.

Target	$\log_2$(#texts)								Method	Reference
[Application]	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$‘	$r=10$		
$(4,3,32)$-SPN	12	28	84	113	124	-	-	-	our	Sect. 5.3
[Serpent]	28	82	113	123	127	-	-	-	degree	[BCC11]

Target	$\log_2$(#texts)								Method	Reference
[Application]	$r=8$	$r=9$	$r=10$	$r=11$	$r=12$	$r=13$	$r=14$	$r=15$		
$(5,2,320)$-SPN	130	258	515	1025	1410	1538	1580	1595	our	Sect. 5.3
[KECCAK-f]	257	513	1025	1409	1537	1579	1593	1598	degree	[BCC11]

AES [U.S01] are classified into $(4,3,16)$- and $(8,7,16)$-SPN, respectively. When we construct the integral distinguisher on (ℓ, d, m)-SPN with 2^D chosen plaintexts, we use a vector $\boldsymbol{k}$ satisfying

$$k_i = \begin{cases} \ell & \text{for } i\ell \le D, \\ D-(i-1)\ell & \text{for } (i-1)\ell \le D < i\ell, \\ 0 & \text{for } D < (i-1)\ell. \end{cases}$$

For the comparison with our integral distinguishers, we first consider the propagation characteristic of the integral property. However, it does not construct a sufficient distinguisher because the P-Layer is any linear function. Next, we estimate the algebraic degree by using the method proposed by Boura et al. We show the method in Appendix A.

As a result, as far as we try, all distinguishers constructed by the division property are "better" than those by previous methods. We summarize integral distinguishers on other (ℓ, d, m)-SPN in Appendix C. We already know the 7-round integral distinguisher on PRESENT in [WW13] and the 4-round integral distinguisher on AES in [KW02]. However, they are constructed by using the specific feature of each block cipher. On the other hand, our method is generic distinguishing attacks against (ℓ, d, m)-SPN. From the result of $(4,3,16)$-SPN, it shows that even if the P-Layer of PRESENT is chosen from any bijective linear functions, the modified PRESENT has the 6-round integral distinguisher. Similarly, from the result of $(8,7,16)$-SPN, it shows that even if the P-Layer of AES is chosen from any bijective linear function, the modified AES still has the 4-round integral distinguisher.

Integral Distinguishers on Serpent and Keccak-f Although our attack is a generic attack, it can create new integral distinguishers on Serpent and KECCAK-f. Serpent is one of AES finalists and is classified into $(4,3,32)$-SPN. The existing integral distinguisher is shown in [ZRHD08], and it shows that Serpent has 3.5-round integral distinguisher. On the other hand, we show that all $(4,3,32)$-SPNs have at least 7-round integral distinguishers

with 2^{124} chosen plaintexts. Table 5 shows the comparison between our distinguishers and previous ones by the degree estimation.

KECCAK is chosen as SHA-3, and the core function KECCAK-f is classified into $(5, 2, 320)$-SPN. Boura et al. estimated the algebraic degree of KECCAK-f in [BCC11]. We search for the integral distinguisher by using Algorithm 2. As a result, our distinguishers can reduce the number of chosen plaintexts compared with previous ones. Table 5 shows the comparison between our distinguishers and previous ones.

6 Toward Dedicated Attack

We introduced the division property in Sect. 3, and proposed distinguishing attacks against the Feistel Network and the SPN in Sect. 4 and Sect. 5, respectively. In this section, we show that the division property is also useful to construct the dedicated attack against specific ciphers. As an example, we show integral distinguishers on AES-like ciphers.

6.1 AES-Like Cipher

(ℓ, d, m)-AES AES is a 128-bit block cipher, and an intermediate text of AES is expressed in a 4×4 matrix whose elements are 8 bits. The round function of AES consists of SubBytes, ShiftRows, MixColumns, and AddRoundKey, where each function is defined as follows:

- SubBytes (SB) : It substitutes each byte in the matrix into another byte by an S-box.
- ShiftRows (SR) : Each byte of the i-th row is rotated $i - 1$ bytes to the left.
- MixColumns (MC) : It diffuses bytes within each column by a linear function.
- AddRoundKey (AK) : A round key is XORed with the intermediate text.

We define an (ℓ, d, m)-AES, where ℓ, d, and m denote the bit length of an S-box, the algebraic degree of an S-box, and the size of the matrix, respectively. This intermediate text is expressed in an $m \times m$ matrix whose elements are ℓ bits. Let $\boldsymbol{X} \in (\mathbb{F}_2^\ell)^{m \times m}$ be an input of the round function, which is arranged as

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,m} \end{bmatrix}.$$

Let $\boldsymbol{Y} \in (\mathbb{F}_2^\ell)^{m \times m}$ be an output of the round function, which is calculated as $\boldsymbol{Y} = (\mathrm{AK} \circ \mathrm{MC} \circ \mathrm{SR} \circ \mathrm{SB})(\boldsymbol{X})$. Each function is the same as that of AES except for the scale. For instance, AES [U.S01] and `LED` [GPPR11] adopt $(8, 7, 4)$-AES and $(4, 3, 4)$-AES, respectively. Moreover, P_{256} of `PHOTON` [GPP11] adopts $(4, 3, 8)$-AES[2].

6.2 Path Search Algorithm for (ℓ, d, m)-AES

Section 5 shows how to construct integral distinguishers on (ℓ, d, m)-SPN, but practical block ciphers have a specific P-Layer. For instance, the P-Layer in AES consists of ShiftRows and MixColumns, and it is not any linear function. Taking into account the structure of the P-Layer, we can construct more effective algorithm. In this section, as an example, we show a path search algorithm to construct integral distinguishers on (ℓ, d, m)-AES.

[2] Since `PHOTON` is a hash function, it uses AddConstant instead of AddRoundKey.

Algorithm 3 Evaluating algorithm against the round function of (ℓ, d, m)-AES

procedure $\texttt{AesFuncEval}(\ell, d, m, \boldsymbol{K})$
  for $r = 1$ to m **do**
    for $c = 1$ to m **do**
      if $k_{r,c} < \ell$ **then** $k_{r,c} \Leftarrow \lceil k_{r,c}/d \rceil$ ▷ SubBytes
      end if
    end for
  end for
  $\boldsymbol{K} \Leftarrow \texttt{ShiftRows}(\boldsymbol{K})$ ▷ ShiftRows
  $k'_c \Leftarrow \sum_{r=1}^{m} k'_{r,c}$ for all c ▷ MixColumns
  $\boldsymbol{k}' \Leftarrow \texttt{sort}(\boldsymbol{k}')$
  return $\boldsymbol{k}'$
end procedure

Algorithm 4 Path search algorithm for integral distinguishers on (ℓ, d, m)-AES

procedure $\texttt{IntegralPathSearch}(\ell, d, m, r = 0, \boldsymbol{K} \in \{0, 1, \ldots, \ell\}^{m \times m})$
  $\boldsymbol{k}^{(1)} \Leftarrow \texttt{AesFuncEval}(\ell, d, m, \boldsymbol{K})$ ▷ 1-st round
  $D \Leftarrow \sum_{c=1}^{m} k_c^{(1)}$
  $q \Leftarrow 1$
  while $1 < D$ **do**
    $r \Leftarrow r + 1$
    for $i = 1$ to q **do**
      $\boldsymbol{K}^{(i,1)}, \ldots, \boldsymbol{K}^{(i,s)} \Leftarrow \texttt{Partition}(\boldsymbol{k}^{(i)})$
      for $j = 1$ to s **do**
        $\bar{\boldsymbol{k}}^{(1)}, \ldots, \bar{\boldsymbol{k}}^{(t)} \Leftarrow \texttt{AesFuncEval}(\ell, d, m, \boldsymbol{K}^{(i,j)})$ ▷ $(r+1)$-th round
        if $(i, j) = (1, 1)$ **then**
          $\boldsymbol{k}'^{(1)}, \ldots, \boldsymbol{k}'^{(q')} \Leftarrow \texttt{SizeReduce}(\bar{\boldsymbol{k}}^{(1)}, \ldots, \bar{\boldsymbol{k}}^{(t)})$
        else
          $\boldsymbol{k}'^{(1)}, \ldots, \boldsymbol{k}'^{(q'')} \Leftarrow \texttt{SizeReduce}(\boldsymbol{k}'^{(1)}, \ldots, \boldsymbol{k}'^{(q')}, \bar{\boldsymbol{k}}^{(1)}, \ldots, \bar{\boldsymbol{k}}^{(t)})$
          $q' \Leftarrow q''$
        end if
      end for
    end for
    $\boldsymbol{k}^{(i)} \Leftarrow \boldsymbol{k}'^{(i)}$ for all $1 \le i \le q'$
    $q \Leftarrow q'$
    $D \Leftarrow \max\{\sum_{c=1}^{m} k_c^{(1)}, \sum_{c=1}^{m} k_c^{(2)}, \ldots, \sum_{c=1}^{m} k_c^{(q)}\}$
  end while
  return r
end procedure

Algorithm 3 evaluates the propagation characteristic of the division property against the round function of AES-like ciphers, and it calls `ShiftRows` and `sort`. `ShiftRows` performs a similar transformation to SR. `sort` is the sorting algorithm, which is useful for feasible implementation. Algorithm 4 shows the path search algorithm, and it calls `Partition`, `AesFuncEval`, and `SizeReduce`. $\texttt{Partition}(\boldsymbol{k}^{(i)})$ calculates all possible $\boldsymbol{K}^{(i,j)}$ satisfying

$$\left(\sum_{r=1}^{m} k_{r,1}^{(i,j)}, \sum_{r=1}^{m} k_{r,2}^{(i,j)}, \ldots, \sum_{r=1}^{m} k_{r,m}^{(i,j)} \right) = (k_1^{(i)}, k_2^{(i)}, \ldots, k_m^{(i)}),$$

where $0 \le k_{r,c}^{(i,j)} \le \ell$ holds. `SizeReduce` eliminates $\boldsymbol{k}^{(i,j)}$ if there exists (i', j') satisfying $\mathbb{S}^{\ell m,m}_{\boldsymbol{k}^{(i,j)}} \subseteq \mathbb{S}^{\ell m,m}_{\boldsymbol{k}^{(i',j')}}$.

Notice that the size of q in the division property extremely expands when the partition of the division property is executed (see the 8-th line in Algorithm 4). Namely, our algorithm

Table 6. The number of chosen plaintexts to construct r-round integral distinguishers on $(4,3,m)$-AES. Our distinguishers are got by implementing Algorithm 2 and Algorithm 4.

Target [Application]	$\log_2$(#texts)						Method	Reference
	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$		
$(4,3,4)$-AES [LED]	4	12	32	52	-	-	our (AES)	Sect. 6.2
	12	28	52	60	-	-	our (SPN)	Sect. 5.3
	28	52	60	63	-	-	degree	[BCC11]
	4	16	-	-	-	-	integral	[DKR97,KW02]
$(4,3,5)$-AES [P_{100} in PHOTON]	4	12	20	72	97	-	our (AES)	Sect. 6.2
	12	28	76	92	-	-	our (SPN)	Sect. 5.3
	28	76	92	98	-	-	degree	[BCC11]
	4	20	-	-	-	-	integral	[DKR97,KW02]
$(4,3,6)$-AES [P_{144} in PHOTON]	4	12	24	84	132	-	our (AES)	Sect. 6.2
	12	28	84	124	140	-	our (SPN)	Sect. 5.3
	28	82	124	138	142	-	degree	[BCC11]
	4	24	-	-	-	-	integral	[DKR97,KW02]
$(4,3,7)$-AES [P_{196} in PHOTON]	4	12	24	84	164	192	our (AES)	Sect. 6.2
	12	28	84	160	184	192	our (SPN)	Sect. 5.3
	28	82	158	184	192	195	degree	[BCC11]
	4	28	-	-	-	-	integral	[DKR97,KW02]
$(4,3,8)$-AES [P_{256} in PHOTON]	4	12	28	92	204	249	our (AES)	Sect. 6.2
	12	28	84	200	237	252	our (SPN)	Sect. 5.3
	28	82	198	237	250	254	degree	[BCC11]
	4	32	-	-	-	-	integral	[DKR97,KW02]

takes large execution time and large memory capacity if we straightforwardly implement our algorithm. Therefore, we use an effective method, which uses the feature of (ℓ,d,m)-AES, for the feasible implementation. Notice that each column of (ℓ,d,m)-AES is equivalent each other. Assuming that the input set has $\mathcal{D}^{\ell m,m}_{\boldsymbol{k},\boldsymbol{k}'}$ that $\boldsymbol{k}'$ is a permutation of elements of $\boldsymbol{k}$, the division property of the next round calculated from $\boldsymbol{k}$ is exactly the same as that from $\boldsymbol{k}'$ because columns of (ℓ,d,m)-AES are equivalent each other. Namely, it is enough to save either, and we implement it by a sorting algorithm (see the 10-th line in Algorithm 3). This technique enables us to execute our path search algorithm feasibly in many parameters.

Results Table 6 shows the number of required chosen plaintexts to construct r-round integral distinguishers on $(4,3,m)$-AES. When we construct the integral distinguisher on (ℓ,d,m)-AES with 2^D chosen plaintexts, we carefully choose the input matrix $\boldsymbol{K}$.

For the comparison with our improved integral distinguishers, we also show integral distinguishers by using the propagation characteristic of the integral property. We also estimate the algebraic degree by the method proposed Boura et al. (see Appendix A). Moreover, since $(4,3,m)$-AES are classified into $(4,3,m^2)$-SPN, we construct integral distinguishers by Algorithm 2.

As a result, as far as we try, all distinguishers constructed by the division property are at least better than those by previous methods. Especially, the advantage of our method is large when we construct the integral distinguisher with the small number of texts. For instance, our method shows that $(4,3,8)$-AES, which is adopted by P_{256} in PHOTON, has the 6-round distinguisher with 2^{92} chosen plaintexts. If we regard $(4,3,8)$-AES as $(4,3,64)$-SPN, 2^{200} chosen plaintexts are required to construct the distinguisher.

7 Conclusions

In this paper, we proposed the fundamental technique to improve integral distinguishers, and showed structural cryptanalyses against the Feistel Network and the SPN. Our new technique uses the division property, which is the generalization of the integral property. It can effectively construct integral distinguishers even if block ciphers have non-bijective functions, bit-oriented structures, and low-degree functions. For the Feistel Network, when the algebraic degree of the F-function is smaller than the bit length of the F-function, our method can attack more rounds than previous generic attacks. Moreover, we theoretically showed that Simon 48, 64, 96, and 128 have 11-, 11-, 13-, and 13-round integral distinguishers, respectively. For the SPN, our method extremely reduces the required number of chosen plaintexts compared with previous methods. Moreover, we improved integral distinguishers on Keccak-f and Serpent. The division property is useful to construct integral distinguishers against specific ciphers. As one example, we showed a path search algorithm to construct integral distinguishers on the AES-like cipher, which is the sub class of the SPN. From this fact, we expect that the division property can construct many improved integral distinguishers against specific ciphers by constructing the dedicated path search algorithm.

Acknowledgments. The authors would like to thank Deukjo Hong for his helpful pointing out.

References

ABB+14. Elena Andreeva, Begül Bilgin, Andrey Bogdanov, Atul Luykx, Florian Mendel, Bart Mennink, Nicky Mouha, Qingju Wang, and Kan Yasuda. PRIMATEs v1.02, 2014. Submission to CAESAR competition.

ABK98. Ross Anderson, Eli Biham, and Lars Knudsen. Serpent: A proposal for the Advanced Encryption Standard. NIST AES Proposal, 1998.

AIK+00. Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita. Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In *SAC*, volume 2012 of *LNCS*, pages 39–56, 2000.

BC13. Christina Boura and Anne Canteaut. On the influence of the algebraic degree of F^{-1} on the algebraic degree of $G \circ F$. *IEEE Transactions on Information Theory*, 59(1):691–702, 2013.

BCC11. Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of Keccak and *Luffa*. In *FSE*, volume 6733 of *LNCS*, pages 252–269, 2011.

BKL+07. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In *CHES*, volume 4727 of *LNCS*, pages 450–466, 2007.

BR03. Paulo S. L. M. Barreto and Vincent Rijmen. The Whirlpool hashing function, 2003. submitted to the NESSIE project, available at `http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html`.

BS01. Alex Biryukov and Adi Shamir. Structural cryptanalysis of SASAS, 2001.

BSS+13. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK families of lightweight block ciphers. *IACR Cryptology ePrint Archive*, 2013:404, 2013.

CSW08. Christophe De Cannière, Hisayoshi Sato, and Dai Watanabe. Hash function *Luffa* - a SHA-3 candidate, 2008. Available at `http://hitachi.com/rd/yrl/crypto/luffa/round1archive/Luffa_Specification.pdf`.

CV02. Anne Canteaut and Marion Videau. Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis. In *EUROCRYPT*, volume 2332 of *LNCS*, pages 518–533, 2002.

DBPA11. Joan Daemen, Guido Bertoni, Michaël Peeters, and Gilles Van Assche. The Keccak reference version 3.0, 2011.

DEMS14. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1, 2014. Submission to CAESAR competition.
DKR97. Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The block cipher Square. In *FSE*, volume 1267 of *LNCS*, pages 149–165, 1997.
DPAR00. Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. The Noekeon block cipher., 2000. submitted to the NESSIE project, available at `http://gro.noekeon.org/`.
DR02. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.
GPP11. Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In *CRYPTO*, volume 6841 of *LNCS*, pages 222–239, 2011.
GPPR11. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In *CHES*, volume 6917 of *LNCS*, pages 326–341, 2011.
IS13. Takanori Isobe and Kyoji Shibutani. Generic key recovery attack on Feistel scheme. In *ASIACRYPT (1)*, volume 8269 of *LNCS*, pages 464–485. Springer, 2013.
KLL+14. Elif Bilge Kavun, Martin Mehl Lauridsen, Gregor Leander, Christian Rechberger, Peter Schwabe, and Tolga Yalçin. Prøst v1.1, 2014. Submission to CAESAR competition.
Knu94. Lars R. Knudsen. Truncated and higher order differentials. In *FSE*, volume 1008 of *LNCS*, pages 196–211, 1994.
Knu02. Lars R. Knudsen. The security of Feistel ciphers with six rounds or less. *J. Cryptology*, 15(3):207–222, 2002.
KW02. Lars R. Knudsen and David Wagner. Integral cryptanalysis (extended abstract). In *FSE*, volume 2365 of *LNCS*, pages 112–127, 2002.
Lai94. Xuejia Lai. Higher order derivatives and differential cryptanalysis. In *Communications and Cryptography*, volume 276 of *The Springer International Series in Engineering and Computer Science*, pages 227–233, 1994.
LR88. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
LWZ11. Yanjun Li, Wenling Wu, and Lei Zhang. Improved integral attacks on reduced-round CLEFIA block cipher. In *WISA*, volume 7115 of *LNCS*, pages 28–39, 2011.
MGH+14. Pawe? Morawiecki, Kris Gaj, Ekawat Homsirikamol, Krystian Matusiewicz, Josef Pieprzyk, Marcin Rogawski, Marian Srebrny, and Marcin Wøjcik. ICEPOLE v1, 2014. Submission to CAESAR competition.
Pat04. Jacques Patarin. Security of random Feistel schemes with 5 or more rounds. In *CRYPTO*, volume 3152 of *LNCS*, pages 106–122, 2004.
SK12. Naoki Shibayama and Toshinobu Kaneko. A peculiar higher order differential of CLEFIA. In *ISITA*, pages 526–530. IEEE, 2012.
STA+14. Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, and Shoichi Hirose. Minalpher v1, 2014. Submission to CAESAR competition.
U.S77. U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology. *DATA ENCRYPTION STANDARD (DES)*, 1977. Federal Information Processing Standards Publication 46.
U.S01. U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology. *Specification for the ADVANCED ENCRYPTION STANDARD (AES)*, 2001. Federal Information Processing Standards Publication 197.
WLV+14. Qingju Wang, Zhiqiang Liu, Kerem Varici, Yu Sasaki, Vincent Rijmen, and Yosuke Todo. Cryptanalysis of reduced-round SIMON32 and SIMON48. In *INDOCRYPT*, volume 8885 of *LNCS*, pages 143–160. Springer, 2014.
WW13. Shengbao Wu and Mingsheng Wang. Integral attacks on reduced-round PRESENT. In *ICICS*, volume 8233 of *LNCS*, pages 331–345, 2013.
WZ11. Wenling Wu and Lei Zhang. LBlock: A lightweight block cipher. In *ACNS*, volume 6715 of *LNCS*, pages 327–344, 2011.
YPK02. Yongjin Yeom, Sangwoo Park, and Iljun Kim. On the security of CAMELLIA against the Square attack. In *FSE*, volume 2365 of *LNCS*, pages 89–99, 2002.
ZRHD08. Muhammad Reza Z'aba, Håvard Raddum, Matthew Henricksen, and Ed Dawson. Bit-pattern based integral attack. In *FSE*, volume 5086 of *LNCS*, pages 363–381, 2008.

A Estimation of Algebraic Degree for (ℓ, d, m)-SPN

If the degree of r iterated round functions is at most D, we can construct the r-round integral distinguisher with 2^{D+1} chosen plaintexts. In a classical method, if the degree of

the round function is at most d, the degree of r iterated round functions is bounded by d^r. In 2011, Boura et al. showed tighter bound as follows.

Theorem 1 ([BCC11]). *Let S be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ corresponding to the concatenation of m smaller S-boxes, defined over $\mathbb{F}_2^{n_0}$. Let δ_k be the maximal degree of the product of any k bits of anyone of these S-boxes. Then, for any function G from $\mathbb{F}_2^n$ into $\mathbb{F}_2$, we have*

$$\deg(G \circ S) \leq n - \frac{n - \deg(G)}{\gamma},$$

where

$$\gamma = \max_{1 \leq i \leq n_0 - 1} \frac{n_0 - i}{n_0 - \delta_i}.$$

By using this bound, we can estimate the degree of (ℓ, d, m)-SPN. For instance, we show the degree of $(4, 3, 64)$-SPN as follows.

Number of rounds	1	2	3	4	5	6	7	8	9
Bound on degree	3	9	27	81	197	236	249	253	255

Therefore, we can construct the 8-round integral distinguisher on $(4, 3, 64)$-SPN with 2^{254} chosen plaintexts.

B Integral Distinguishers on (ℓ, d)-Feistel

Table 7 shows integral distinguishers on (ℓ, d)-Feistel, where (ℓ, d)-Feistel is defined in Sect. 4.1. If we construct the dedicated path search algorithm for the specific cipher, we expect that the algorithm can create better integral distinguishers.

Table 7. The number of required chosen plaintexts to construct r-round integral distinguishers on (ℓ, d)-Feistel. We get these values by implementing Algorithm 1.

Target	F-function	$\log_2$(#texts)									Examples
		$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	$r=11$	$r=12$	$r=13$	$r=14$	
$(16,2)$	non-bijection	17	25	29	31	-	-	-	-	-	Simon 32 [BSS+13]
	bijection	16	23	28	30	31	-	-	-	-	
$(24,2)$	non-bijection	17	29	39	44	46	47	-	-	-	Simon 48 [BSS+13]
	bijection	17	27	38	43	46	47	-	-	-	
$(32,2)$	non-bijection	17	33	49	57	61	63	-	-	-	Simon 64 [BSS+13]
	bijection	17	32	47	56	60	62	63	-	-	
$(48,2)$	non-bijection	17	33	57	77	87	92	94	95	-	Simon 96 [BSS+13]
	bijection	17	33	55	76	86	91	94	95	-	
$(64,2)$	non-bijection	17	33	65	97	113	121	125	127	-	Simon 128 [BSS+13]
	bijection	17	33	64	95	112	120	124	126	127	

Target	F-function	$\log_2$(#texts)									Examples
		$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	$r=11$	
$(32,5)$	non-bijection	6	26	51	62	-	-	-	-	-	DES [U.S77]
	bijection	6	26	46	61	-	-	-	-	-	
$(48,5)$	non-bijection	6	26	64	90	95	-	-	-	-	
	bijection	6	26	59	89	95	-	-	-	-	
$(64,5)$	non-bijection	6	26	77	118	126	-	-	-	-	
	bijection	6	26	72	117	126	-	-	-	-	

Target	F-function	$\log_2$(#texts)									Examples
		$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	$r=11$	
$(32,7)$	non-bijection	8	35	60	-	-	-	-	-	-	
	bijection	8	32	59	-	-	-	-	-	-	
$(48,7)$	non-bijection	8	49	90	-	-	-	-	-	-	
	bijection	8	48	84	95	-	-	-	-	-	
$(64,7)$	non-bijection	8	50	104	125	-	-	-	-	-	
	bijection	8	50	98	124	-	-	-	-	-	Camellia [AIK+00]

Target	F-function	$\log_2$(#texts)									Examples
		$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	$r=11$	
$(32,31)$	non-bijection	32	62	-	-	-	-	-	-	-	
	bijection	32	32	63	-	-	-	-	-	-	
$(48,47)$	non-bijection	48	94	-	-	-	-	-	-	-	
	bijection	48	48	95	-	-	-	-	-	-	
$(64,63)$	non-bijection	64	126	-	-	-	-	-	-	-	
	bijection	64	64	127	-	-	-	-	-	-	
$(32,32)$	non-bijection	33	-	-	-	-	-	-	-	-	
$(48,48)$	non-bijection	49	-	-	-	-	-	-	-	-	
$(64,64)$	non-bijection	65	-	-	-	-	-	-	-	-	

C Integral Distinguishers on (ℓ, d, m)-SPN

Table 8 shows integral distinguishers on (ℓ, d, m)-SPN, where (ℓ, d, m)-SPN is defined in Sect. 5.1. If we construct the dedicated path search algorithm for the specific cipher, we expect that the algorithm can create better integral distinguishers.

Table 8. The number of required chosen plaintexts to construct r-round integral distinguishers on (ℓ, d, m)-SPN. We get these values by implementing Algorithm 2.

Target	Size	$\log_2$(#texts)							Examples
	(bits)	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	
$(4,3,16)$	64	28	52	60	-	-	-	-	PRESENT [BKL+07], LED [GPPR11]
$(4,3,24)$	96	28	76	89	-	-	-	-	
$(4,3,32)$	128	28	84	113	124	-	-	-	Serpent [ABK98], NOEKEON [DPAR00]
$(4,3,40)$	160	28	84	136	152	-	-	-	
$(4,3,48)$	192	28	84	156	180	188	-	-	
$(4,3,56)$	224	28	84	177	209	220	-	-	
$(4,3,64)$	256	28	84	200	237	252	-	-	Prøst-128 [KLL+14], Minalpher-P [STA+14]
$(4,3,128)$	512	28	84	244	424	484	504	509	Prøst-256 [KLL+14]

Target	Size	$\log_2$(#texts)							Examples
	(bits)	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$	$r=11$	
$(5,2,40)$	200	18	35	65	130	178	195	-	PRIMATE-80 [ABB+14]
$(5,2,56)$	280	18	35	65	130	230	265	275	PRIMATE-120 [ABB+14]
$(5,2,64)$	320	18	35	65	130	258	300	315	ASCON Permutation [DEMS14]

Target	Size	$\log_2$(#texts)							Examples
	(bits)	$r=9$	$r=10$	$r=11$	$r=12$	$r=13$	$r=14$	$r=15$	
$(5,2,160)$	800	258	515	705	770	790	798	-	KECCAK-f[800] [DBPA11]
$(5,2,256)$	1280	258	515	1025	1195	1253	1271	1278	
$(5,2,320)$	1600	258	515	1025	1410	1538	1580	1595	KECCAK-f[1600] [DBPA11]

Target	Size	$\log_2$(#texts)							Examples
	(bits)	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	
$(5,4,40)$	200	20	65	170	195	-	-	-	
$(5,4,56)$	280	20	65	230	270	-	-	-	
$(5,4,64)$	320	20	65	260	305	-	-	-	
$(5,4,160)$	800	20	65	260	665	770	795	-	
$(5,4,256)$	1280	20	65	260	1025	1220	1265	-	ICEPOLE Permutation [MGH+14]
$(5,4,320)$	1600	20	65	260	1025	1460	1565	1595	

Target	Size	$\log_2$(#texts)							Examples
	(bits)	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	
$(8,7,16)$	128	56	120	-	-	-	-	-	AES [U.S01]
$(8,7,24)$	192	56	176	-	-	-	-	-	Rijndael-192 [DR02]
$(8,7,32)$	256	56	232	-	-	-	-	-	Rijndael-256 [DR02]
$(8,7,64)$	512	56	344	488	-	-	-	-	WHIRLPOOL primitive [BR03]