IEICE TRANS. FUNDAMENTALS, VOL.E107–A, NO.8 AUGUST 2024

PAPER

SAT-Based Analysis of Related-Key Impossible Distinguishers on Piccolo and (Tweakable) TWINE

Shion UTSUMI[†a)], ***Nonmember***, **Kosei SAKAMOTO**[†,††], ***and*** **Takanori ISOBE**[†], ***Members***

SUMMARY Lightweight block ciphers have gained attention in recent years due to the increasing demand for sensor nodes, RFID tags, and various applications. In such a situation, lightweight block ciphers Piccolo and TWINE have been proposed. Both Piccolo and TWINE are designed based on the Generalized Feistel Structure. However, it is crucial to address the potential vulnerability of these structures to the impossible differential attack. Therefore, detailed security evaluations against this attack are essential. This paper focuses on conducting bit-level evaluations of Piccolo and TWINE against related-key impossible differential attacks by leveraging SAT-aided approaches. We search for the longest distinguishers under the condition that the Hamming weight of the active bits of the input, which includes plaintext and master key differences, and output differences is set to 1, respectively. Additionally, for Tweakable TWINE, we search for the longest distinguishers under the related-tweak and related-tweak-key settings. The result for Piccolo with a 128-bit key, we identify the longest 16-round distinguishers for the first time. In addition, we also demonstrate the ability to extend these distinguishers to 17 rounds by taking into account the cancellation of the round key and plaintext difference. Regarding evaluations of TWINE with a 128-bit key, we search for the first time and reveal the distinguishers up to 19 rounds. For the search for Tweakable TWINE, we evaluate under the related-tweak-key setting for the first time and reveal the distinguishers up to 18 rounds for 80-bit key and 19 rounds for 128-bit key.

key words: *Piccolo, TWINE, related-key impossible differential attack, SAT*

1. Introduction

1.1 Background

In recent years, there has been an increasing opportunity for real-world data, such as in healthcare, manufacturing, and automotive sectors, to be exchanged in the digital world. These systems often incorporate resource-constrained RFID tags or sensor nodes, and traditional encryption methods have been unable to provide security for these devices. Therefore, in such situations, lightweight ciphers that can replace conventional block ciphers have gained attention in recent years.

Piccolo [1], a lightweight block cipher that can be implemented in hardware with a small circuit size, was proposed at CHES 2011. TWINE [2], a lightweight block cipher that is designed well-balanced for both hardware and software, was proposed at SAC 2012. Both lightweight block ciphers feature a well-designed Generalized Feistel Structure (GFS) to improve diffusion properties. However, one of the most effective attacks on Piccolo in both single-key and related-key settings are the impossible differential attacks by Azimi et al. [3] and Minier [4], respectively. Regarding TWINE, impossible differential attacks by Zheng et al. [5] and Biryukov et al. [6] are effective under the single-key setting for 80-bit and 128-bit keys, respctively. In the related-key setting, the attack by Wei et al. [7] indicate a 24-round impossible differential attack on TWINE with a 80-bit key. Thus, impossible differential attacks on these ciphers may harbor hidden vulnerabilitiy and should be investigated in more detail.

Traditionally, evaluations against impossible differentials have been derived from manual calculations using the miss-in-the-middle method [8]. Subsequently, various automated search methods have been developed, including the '$\mathcal{U}$-method' by Kim et al. [9] and its improvements [10], and the constraint programming (CP)-aided method proposed by Sun et al. [11] and others. These methods rely on truncated differential-based evaluations and do not consider the differential propagation within the internals of S-boxes. Later on, Sasaki and Todo introduced a mixed integer linear programming (MILP)-aided method for searching impossible differentials [12]. This method transforms the previous MILP-aided differential search into an impossible differential search, making it possible to consider the differential property within the S-box.

On the other hand, cryptanalyses using automated tools are also conducted with SAT solvers. In recent years, Sun et al. have proposed an evaluation method against differential and linear cryptanalysis [13] that enables faster evaluation than MILP-aided methods. This faster method enables the discovery of bit-level differentials and linear characteristics over a larger number of rounds. The SAT-aided approach has the potential to lead to the discovery of more rigorous characteristics over a higher number of rounds, also about impossible differential evaluations.

In this paper, we mainly focus on related-key, related-tweak, and related-tweak-key impossible distinguishers as it is important to understand their structure property and the security of underlying components as pseudo random permutations. However, due to the large input space and high computational cost, the bit-level searches for impossible differential distinguishers under these settings have been challenging. Therefore, in this study, we attempt to achieve distinguisher searches by utilizing a SAT-aided efficient evaluation method.

Manuscript received November 20, 2023.
Manuscript publicized February 8, 2024.
[†]Univercity of Hyogo, Kobe-shi, 650-0047 Japan.
[††]Mitsubishi Electric Corporation, Kanagawa, 247-8051 Japan.
a) E-mail: shioneutsumi0705@gmail.com
DOI: 10.1587/transfun.2023EAP1149

1.2 Our Contribution

In this paper, we conduct bit-level analyses of related-key impossible differential attacks on Piccolo and (Tweakable) TWINE by converting SAT-aided approaches for differential cryptanalysis proposed by Sun et al. [13]. As for the evaluation of Piccolo, since the round keys are linearly determined in the key scheduling by the round that initiates the encryption, we conduct evaluations by varying the round at which encryption starts. Additionally, for Tweakable TWINE, we search for the first time under the related-tweak-key setting, considering differences not only in the plaintext and key but also in the tweak, using the same method.

In the differential search with SAT solver, we represent the differential propagation at each round of the cipher using Conjunctive Normal Form (CNF)-format constraints. Additionally, we utilize auxiliary variables to express the differential probabilities in the non-linear functions and calculate the differential characteristic probabilities using an objective function based on cardinality constraints.

In this evaluation against impossible differentials, to examine the impossibility of given input and output, we simply add constraints to fix the input and output differences without object function. Moreover, searching through all possible input-output pairs is computationally challenging. Therefore, we restrict the search under the condition that the Hamming weight of the active bits of the input, which includes plaintext and master key differences, and output differences is 1, respectively.

Furthermore, by taking into account the cancellation of differences between the key and data processing parts through manual calculation, we successfully extend the impossible differentials of Piccolo with a 128-bit key by one round based on the results obtained by the SAT-aided evaluation.

A summary of our results is shown in Table 1. For the first time, we succeed in identifying the longest 16-round distinguishers for Piccolo with a 128-bit key using the SAT-aided method. In addition, considering the cancellation of the round key and plaintext difference, we extend the distinguishers to 17 rounds. Regarding the evaluation of TWINE with a 128-bit key, we search for the first time and reveal the distinguishers up to 19 rounds. For the search for Tweakable TWINE, we conduct the evaluation under the related-tweak-key setting for the first time and revealed the distinguishers up to 18 rounds for 80-bit key and 19 rounds for 128-bit key.

Table 1 The summary of the longest related-key Impossible distinguishers for each target.

Target	Key length	Existing	Ours
Piccolo	80 bit	10R[4]	10R
	128 bit	12R[4]	17R
TWINE	80 bit	15R[7]	15R
	128 bit	None	19R
T-TWINE	80 bit (related-tweak)	18R[14]	18R
	80 bit (related-tweak-key)	None	18R
	128 bit (related-tweak)	18R[14]	18R
	128 bit (related-tweak-key)	None	19R

1.3 Organization

This paper is organized as follows. We first describe the related-key impossible differential attack and a brief explanation of security evaluation by SAT solvers in Sect. 2. In Sect. 3, we describe the specifications of Piccolo, TWINE, and Tweakable TWINE. In Sect. 4, we explain the specific security evaluation methods against related-key impossible distinguisher attacks using SAT solvers. We show the search result of the related-key impossible distinguishers in Sect. 5 and present some considerations in Sect. 6. Finally, Sect. 7 concludes this paper.

2. Preliminaries

In this section, we describe the related-key impossible attack and a brief explanation of the security evaluation using SAT solvers.

2.1 Related-Key Impossible Differential Attack

Related-key impossible differential attack was first proposed in 2003 by Jakimoski et al. [15] as an attack combining an impossible differential attack and a related-key differential attack.

The impossible differential attack was independently introduced by Biham et al. in 1998 [16] and Knudsen in 1999 [17]. Impossible differential attack is one of the most powerful attacks against block ciphers based on Generalized Feistel structure (GFS). In contrast to the differential attack, the impossible differential attack searches for an output differential that propagates with a probability of 0 for a given input differential. When this input-output differential pair is found in r rounds, it is called the impossible distinguisher in r rounds, and the disginguisher attack is successful.

The related-key differential attack was introduced by Biham in 1994 [18] allow an attacker to know some relations between different keys without knowing the keys themselves and to cipher under those keys some plaintext. This is a highly disadvantageous attack condition for cryptographic designers, but it allows for evaluating the security against attacks exploiting weak key schedule functions.

In the related-key impossible differential attack, the attacker introduces differences in the keys and produces an impossible differential.

2.2 Security Evaluation of Block Cipher by SAT Solver

In recent years, solver-aided automatic search methods greatly contributed to the cryptanalysis of symmetric-key primitives. The impactful solver-aided search began with the method using mixed integer linear programming (MILP) by Mouha et al. [19], and it has been employed in various

sides of symmetric-key primitives [20], [21]. Subsequently, the search method utilizing the Boolean Satisfiability Problem (SAT) proposed by Sun et al. [13] has been suggested as a more efficient approach compared to MILP. Since the introduction of SAT-aided evaluation methods, various contributions have emerged [22], [23]. The SAT-aided automatic evaluation method for symmetric key block ciphers is currently one of the effective evaluation methods, and in this study, we employ "parkissat†" as a SAT solver.

SAT is concerned with determining whether a given Boolean formula can be evaluated to True by assigning appropriate Boolean values to its variables. To construct a SAT model for evaluation, the method expresses all operations in a cryptographic scheme as Conjunctive Normal Form (CNF) and assigns them to a SAT model as constraints. The evaluation method used in this research is described in Sect. 4.

3. Specification of Our Target

In this section, we describe the specification of lightweight block ciphers Piccolo and (Tweakable) TWINE.

3.1 Notation

The notation in this paper is shown below.

- $a_{(b)}$: the bit length of b
- $a|b$ or $(a|b)$: concatenate a and b
- $a \leftarrow b$: update value a to b
- ${}^t\boldsymbol{a}$: transpose $\boldsymbol{a}$ vector or matrix
- $\{a\}_b$: express a as a radix of b
- a^L, a^R : left and right half of a, respectively
- $F(a)$: input a to the F-function
- $RP(a)$: permutate a with round permutation
- $Rotz(a)$: the z-bit left cyclic shift of a
- P, C : represent the plaintext and the ciphertext, respectively
- K, T : represent the master key and the tweak, respectively

3.2 Description of Piccolo

Piccolo [1] is a lightweight block cipher that was proposed at CHES 2011, and is capable of being implemented in hardware with a low circuit size. Piccolo is designed based on the GFS and has a block length of 64 bits, supporting 80-bit and 128-bit key lengths. If the key length needs to be specified, we write Piccolo-80 or Piccolo-128 to denote the corresponding version. The number of rounds varies depending on the key length, with 25 or 31 rounds for 80 or 128 bits, respectively. However, both variants of Piccolo share similar processes in key scheduling and data processing. The encryption function of Piccolo is depicted in Fig. 1. The following presents detailed specifications for Piccolo.

†https://github.com/songfu1983/ParKissat-RS

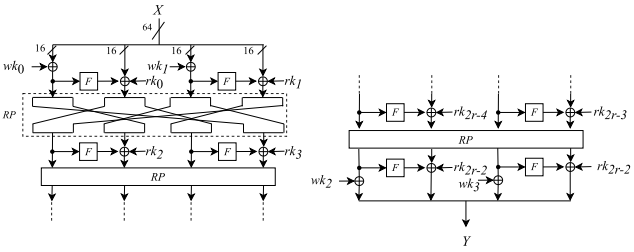

Fig. 1 Data encryption of Piccolo.

Algorithm 1 Piccolo algorithm.

$X_{0(16)}|X_{1(16)}|X_{2(16)}|X_{3(16)} \leftarrow P_{(64)}$
$X_0 \leftarrow X_0 \oplus wk_0$
$X_2 \leftarrow X_2 \oplus wk_1$
for $i \leftarrow 0$ to (24 or 30) **do**
  $X_1 \leftarrow X_1 \oplus F(X_0) \oplus rk_{2i}$
  $X_3 \leftarrow X_3 \oplus F(X_2) \oplus rk_{2i+1}$
  $X_0|X_1|X_2|X_3 \leftarrow RP(X_0|X_1|X_2|X_3)$
end for
$X_1 \leftarrow X_1 \oplus F(X_0) \oplus rk_{2r-2}$
$X_3 \leftarrow X_3 \oplus F(X_2) \oplus rk_{2r-1}$
$X_0 \leftarrow X_0 \oplus wk_2$
$X_2 \leftarrow X_2 \oplus wk_3$
$C_{(64)} \leftarrow X_0|X_1|X_2|X_3$

3.2.1 Data Processing Part

In the data processing part, the 64-bit input is first divided into four 16-bit segments, and then the encryption is performed using a 4-line GFS. Before the first round and after the last round, the pre- and post-whitening keys are added. The encryption algorithm is presented in Algorithm 1.

The F-function consists of two S-box layers separated by a diffusion matrix M. Each S-box layer applies the same four 4×4-bit bijective S-boxes presented in Table 2 in parallel. The diffusion function updates the internal state by the matrix M as follows:

$$(x_{0(4)}, x_{1(4)}, x_{2(4)}, x_{3(4)})^T = M \cdot (x_{0(4)}, x_{1(4)}, x_{2(4)}, x_{3(4)})^T.$$

where the multiplication is performed over a finite field GF (2^4). The round permutation RP acts at byte level and permutes the bytes of the current block as follows:

$$\begin{aligned}\mathrm{RP}:&(x_{0(8)}, x_{1(8)} \ldots x_{7(8)})\\ &\rightarrow (x_{2(8)}, x_{7(8)}, x_{4(8)}, x_{1(8)}, x_{6(8)}, x_{3(8)}, x_{0(8)}, x_{5(8)}).\end{aligned}$$

3.2.2 Key Scheduling Part

Piccolo supports 80-bit and 128-bit keys. We first define 16-bit constants con_i^{80} and con_i^{128}, and then we describe the process of key scheduling part.

Constant Values
The constants con_i^{80} for Piccolo-80 and con_i^{128} for Piccolo-128 are generated as follows:

Table 2 S-box of Piccolo.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S[x]$	e	4	b	2	3	8	0	9	1	a	7	f	6	c	5	d

Algorithm 2 Key Schedule function for Piccolo-80.

$k_{0(16)}|k_{1(16)}|k_{2(16)}|k_{3(16)}|k_{4(16)} \leftarrow K_{(80)}$
$wk_0 \leftarrow k_0^L|k_1^R, wk_1 \leftarrow k_1^L|k_0^R, wk_2 \leftarrow k_4^L|k_3^R, wk_3 \leftarrow k_3^L|k_4^R$
for $i \leftarrow 0$ to $(2r-1)$ **do**

$$(rk_{2i}, rk_{2i+1}) \leftarrow (con_{2i}^{80}, con_{2i+1}^{80}) \oplus \begin{cases} (k_2, k_3) \text{ (if } i \bmod 5 = 0 \text{ or } 2) \\ (k_0, k_1) \text{ (if } i \bmod 5 = 1 \text{ or } 4) \\ (k_4, k_4) \text{ (if } i \bmod 5 = 3) \end{cases}$$

end for

Algorithm 3 Key Schedule function for Piccolo-128.

$k_{0(16)}|k_{1(16)}|k_{2(16)}|k_{3(16)}|k_{4(16)}|k_{5(16)}|k_{6(16)}|k_{7(16)} \leftarrow K_{(128)}$
$wk_0 \leftarrow k_0^L|k_1^R, wk_1 \leftarrow k_1^L|k_0^R, wk_2 \leftarrow k_4^L|k_7^R, wk_3 \leftarrow k_7^L|k_4^R$
for $i \leftarrow 0$ to $(2r-1)$ **do**
  if $(i+2) \bmod 8 = 0$ **then**
    $(k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7) \leftarrow (k_2, k_1, k_6, k_7, k_0, k_3, k_4, k_5)$
  end if
  $rk_i \leftarrow k_{(i+2) \bmod 8} \oplus con_i^{128}$
end for

$$\begin{cases} (con_{2i}^{80}|con_{2i+1}^{80}) \leftarrow (c_{i+1}|c_0|c_{i+1}|00_2|c_{i+1}|c_0|c_{i+1}) \oplus \{0f1e2d3c\}_{16}, \\ (con_{2i}^{128}|con_{2i+1}^{128}) \leftarrow (c_{i+1}|c_0|c_{i+1}|00_2|c_{i+1}|c_0|c_{i+1}) \oplus \{6547a98b\}_{16}, \end{cases}$$

where c_i is a 5-bit representation of i, e.g., $c_{11} = \{01011\}_2$.

Key Schedule for Piccolo-80
The key scheduling function for Piccolo-80 provides the whitening keys $wk_{i(16)}(0 \leq i < 4)$ and the round keys $rk_{j(16)}(0 \leq j < 2r)$ for the data processing part from a 80-bit master key K as following Algorithm 2.

Key Schedule for Piccolo-128
The key scheduling function for Piccolo-128 provides the whitening keys $wk_{i(16)}(0 \leq i < 4)$ and the round keys $rk_{j(16)}(0 \leq j < 2r)$ for the data processing part from a 128-bit master key K as following Algorithm 3.

3.3 Description of (Tweakable) TWINE

TWINE [2] is a lightweight block cipher proposed by Suzaki et al. It realizes efficient implementation on both quite small hardware and microcontrollers. TWINE employs a GFS and has a block length of 64 bits, supporting 80-bit and 128-bit key lengths. We denote these versions of TWINE as TWINE-80 and TWINE-128, respectively. TWINE-80 and TWINE-128 employs the same 36-round round function and a similar key schedule function. Furthermore, Tweakable TWINE incorporates the same structure as TWINE with the addition of a tweak schedule. In the following, we denote each key length of Tweakable TWINE as T-TWINE-80 and T-TWINE-128. The encryption function of TWINE and T-TWINE is depicted in Fig. 2. The following presents detailed specifications for T-TWINE.

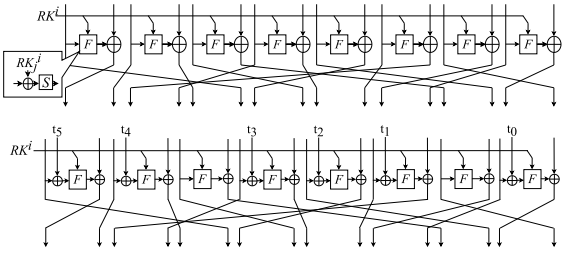

Fig. 2 Data encryption of TWINE (top) and T-TWINE (bottom).

3.3.1 Data Processing Part

In the data processing part, the 64-bit plaintext is divided into 16 4-bit segments, and then the encryption is performed using a 16-line GFS. Each Feistel function simply consist of xor with round keys and mapping using the S-box presented in Table 3. After the above operation, the advanced shuffle selected to improve the diffusion is applied according to the following:

$$\begin{aligned}&(x_{0(4)}, x_{1(4)} \ldots x_{15(4)}) \\ &\quad \rightarrow (x_{5(4)}, x_{0(4)}, x_{1(4)}, x_{4(4)}, x_{7(4)}, x_{12(4)}, x_{3(4)}, x_{8(4)}, \\ &\qquad x_{13(4)}, x_{6(4)}, x_{9(4)}, x_{2(4)}, x_{15(4)}, x_{10(4)}, x_{11(4)}, x_{14(4)}).\end{aligned}$$

In the case of T-TWINE, as shown in Fig. 2, the round tweak is added designated branches.

3.3.2 Key Schduling Part

TWINE supports 80-bit and 128-bit keys. We first define 6-bit constants $CON_{i(6)}$, and then we describe the process of key scheduling part.

Constant Values
The 6-bit round constants, $CON_{i(6)} = CON_{i(3)}^L || CON_{i(3)}^R$, are defined as 2^i in GF(2^6) with primitive polynomial $z^6 + z + 1$.

Key Schedule for TWINE-80
The key scheduling function for TWINE-80 provides the 36 round keys $rk_{i(32)}(0 \leq i < 36)$ from a 80-bit master key K for the data processing part as following Algorithm 4.

Key Schedule for TWINE-128
The key scheduling function for TWINE-128 provides the 36 round keys $rk_{i(32)}(0 \leq i < 36)$ from a 128-bit master key K for the data processing part as following Algorithm 5.

3.3.3 Tweak Scheduling Part

The tweak scheduling function of T-TWINE provides the 24-

Table 3 S-box of TWINE.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S[x]$	c	0	f	a	2	b	9	5	8	3	d	7	1	e	6	4

Algorithm 4 Key Schedule function for TWINE-80.

$k_{0(4)}|k_{1(4)}|\ldots|k_{18(4)}|k_{19(4)} \leftarrow K_{(80)}$
for $i \leftarrow 0$ to 34 **do**
  $rk_i \leftarrow k_1|k_3|k_4|k_6|k_{13}|k_{14}|k_{15}|k_{16}$
  $k_1 \leftarrow k_1 \oplus S(k_0),\ k_4 \leftarrow k_4 \oplus S(k_{16})$
  $k_7 \leftarrow k_7 \oplus 0|CON^L_{i+1},\ k_7 \leftarrow k_{19} \oplus 0|CON^R_{i+1}$
  $k_0|\cdots|k_3 \leftarrow Rot4(k_0|\cdots|k_3)$
  $k_0|\cdots|k_{19} \leftarrow Rot16(k_0|\cdots|k_{19})$
end for
$rk_{35} \leftarrow k_1|k_3|k_4|k_6|k_{13}|k_{14}|k_{15}|k_{16}$
$rk_{(32\times36)} \leftarrow rk_{0(32)}|rk_{2(32)}|\ldots|rk_{34(32)}|rk_{35(32)}$

Algorithm 5 Key Schedule function for TWINE-128.

$k_{0(4)}|k_{1(4)}|\ldots|k_{18(4)}|k_{31(4)} \leftarrow K_{(128)}$
for $i \leftarrow 0$ to 34 **do**
  $rk_i \leftarrow k_2|k_3|k_{12}|k_{15}|k_{17}|k_{18}|k_{28}|k_{31}$
  $k_1 \leftarrow k_1 \oplus S(k_0),\ k_4 \leftarrow k_4 \oplus S(k_{16}),\ k_{23} \leftarrow k_{23} \oplus S(k_{30})$
  $k_7 \leftarrow k_7 \oplus 0|CON^L_{i+1},\ k_7 \leftarrow k_{19} \oplus 0|CON^R_{i+1}$
  $k_0|\cdots|k_3 \leftarrow Rot4(k_0|\cdots|k_3)$
  $k_0|\cdots|k_{31} \leftarrow Rot16(k_0|\cdots|k_{31})$
end for
$rk_{35} \leftarrow k_2|k_3|k_{12}|k_{15}|k_{17}|k_{18}|k_{28}|k_{31}$
$rk_{(32\times36)} \leftarrow rk_{0(32)}|rk_{2(32)}|\ldots|rk_{34(32)}|rk_{35(32)}$

Algorithm 6 Tweak scheduling function of T-TWINE

$t^1_{0(4)}|t^1_{1(4)}|\cdots|t^1_{15(4)} \leftarrow T_{(64)}$
for $i \leftarrow 0$ to 35 **do**
  $RT^i \leftarrow t^i_{0(4)}|t^i_{1(4)}|t^i_{2(4)}|t^i_{3(4)}|t^i_{4(4)}|t^i_{5(4)}|$
  for $h = 0$ to 5 **do**
    $t^i_{\pi^t[h]} \leftarrow t^i_h$
  end for
  for $h = 0$ to 15 **do**
    $t^{i+1}_{(h-6)\bmod16} \leftarrow t^i_h$
  end for
end for
$TK_{(24\times36)} \leftarrow RT^0|RT^2|\cdots|RT^{35}$

bit 36 round tweaks $TK_i (0 \le i < 36)$ for the data processing part from a 64-bit tweak T input as following Algorithm 6.

4. Evaluation Method of Using SAT Solver

In this paper, we conduct bit-level evaluations using the SAT-aided method for the related-key impossible differential attack. The evaluations are mainly based on the method proposed by Sun et al. [13]. We explain the modeling method for the SAT model for the evaluations of the related-key impossible differential attack below.

4.1 SAT Model for Differential Propagation

When using SAT solvers to search for differential distinguishers in a cipher that includes various operations such as linear and nonlinear layers, it is necessary to convert the relationship between operations and differentials of the cipher into a CNF-formatted SAT problem.

The clauses in a CNF formula regarding the search for the optimal differential are classified into two groups. The first group represents the propagations of differences inside the cipher, and the second one measures the non-random feature of the trail, which can be set as the number of active S-boxes, the differential probability optionally. However, in the evaluation of impossible differentials, since the number of active S-box or differential probabilities is not taken into account, the CNF of boolean cardinality constraints for the object function of calculating number of active S-boxes or differential probability is not used. Here, we explain the SAT models of several operations used to discover impossible differential distinguishers.

Differential Model 1 (Branching) [13]. *Let $y = f(x)$ be a branching function, where $x \in \mathbb{F}_2$ is the input variable, and the output variables $y = (y_0, y_1, \ldots, y_{n-1}) \in \mathbb{F}_2^n$ is caluculated as $y_0 = y_1 = \cdots = y_{n-1} = x$. Then, $(\alpha, \beta_0, \beta_1, \ldots, \beta_{n-1})$ is a valid differential trail of function if and only if it satisfies all the equations in the following:*

$$\left.\begin{array}{l} \alpha \vee \overline{\beta_i} = 1 \\ \overline{\alpha} \vee \beta_i = 1 \end{array}\right\}, 0 \le i \le n-1.$$

Differential Model 2 (XOR) [13]. *Let $y = f(x)$ be a function compressed by an XOR, where $x = (x_0, x_1, \ldots, x_{n-1}) \in \mathbb{F}_2^n$ is the input variables, and the output variable $y \in \mathbb{F}_2$ is calculated as $y = x_0 \oplus x_1 \oplus \cdots \oplus x_{n-1}$.*
When $n = 2$, $(\alpha_0, \alpha_1, \beta)$ is a valid differential trail of function if and only if it satisfies all the equations in the following:

$$\left.\begin{array}{l} \alpha_0 \vee \alpha_1 \vee \overline{\beta} = 1 \\ \alpha_0 \vee \overline{\alpha_1} \vee \beta = 1 \\ \overline{\alpha_0} \vee \alpha_1 \vee \beta = 1 \\ \overline{\alpha_0} \vee \overline{\alpha_1} \vee \overline{\beta} = 1 \end{array}\right\}.$$

When $n \ge 3$, let A be the set $\{(\alpha_0, \alpha_1, \ldots, \alpha_n) \in \mathbb{F}_2^{n+1} | \alpha_0 \oplus \alpha_1 \oplus \ldots \oplus \alpha_n = 1\}$. Then the differential trail $(\alpha_0, \alpha_1, \ldots, \alpha_{n-1}, \beta)$ is valid if and only if it satisfies all the following equations:

$$(\alpha_0 \oplus a_0) \vee (\alpha_1 \oplus a_1) \vee \cdots \vee (\alpha_{n-1} \oplus a_{n-1}) \vee (\beta \oplus a_n) = 1,$$
$$(a_0, a_1, \ldots, a_n) \in A.$$

Differential Model 3 (S-box). While we primarily represent differential propagation of S-boxes based on the method proposed by Sun et al. [13], here, we represent the differential

propagation without auxiliary variables.

Let $Pr(\alpha,\beta)$ be the probability of the difference for an n-bit input-output S-box, where $\alpha \in \mathbb{F}_2^n$ is the input difference and $\beta \in \mathbb{F}_2^n$ is the output difference. To create a differential model for the S-box in SAT, we prepare the following Boolean functions:

$$g(\alpha,\beta) = \begin{cases} 0, & \text{if } Pr(\alpha,\beta) = 0 \\ 1, & \text{otherwise.} \end{cases}$$

This represents whether the variable assignment for input/output differences in the S-box is valid or not. We define the set of invalid variable assignments for the S-box, where $g(\alpha,\beta) = 0$, as set A.

$$A = \{(a,b) \in \mathbb{F}_2^{2n} | g(a,b) = 0\}$$

To construct the model of the S-box, each invalid assignment of the set A can be excluded from the set $\mathbb{F}_2^{2n}$ by the following equation, where $|A|$ is the number of invalid assignments.

$$\bigvee_{i=0}^{n-1}(\alpha_i \oplus a_i^l) \vee \bigvee_{j=0}^{n-1}(\beta_j \oplus b_j^l) = 1,$$
$$(a^l, b^l) \in A, \quad 0 \le l \le |A| - 1.$$

The Boolean function representing the differential propagation of the S-box is expressed as follows by excluding all invalid assignments from the solution space.

$$h(\alpha,\beta) = \bigwedge_{l=0}^{|A|-1}\left(\bigvee_{i=0}^{n-1}(\alpha_i \oplus a_i^l) \vee \bigvee_{j=0}^{n-1}(\beta_j \oplus b_j^l)\right) = 1.$$

The above equation is equivalent to the following equation, and the minimum number of clauses can be constructed using software such as Logic Friday [†].

$$h(\alpha,\beta) = \bigwedge_{(a,b)\in\mathbb{F}_2^{2n}}\left(g(a,b) \vee \bigvee_{i=0}^{n-1}(\alpha_i \oplus a_i^l) \vee \bigvee_{j=0}^{n-1}(\beta_j \oplus b_j^l)\right).$$

4.2 Related-Key Impossible Differential Evaluation Model

For the evaluation of the impossible differential attack, we conduct modeling of the differential propagation of the encryption algorithm and verify the validity of the propagation of given input differences and output differences. In the evaluation of the related-key impossible differentials, we also model the differential propagation of the key schedule function and then searched for distinguishers by constraining input-output patterns to evaluate whether the differential propagations of these patterns are possible. We impose constraints such that the Hamming weight of the active difference in the input is 1 for plaintext and key (and tweak), and the Hamming weight of the active difference in the output is 1 for the ciphertext.

Let α denote the input difference of n bits including the key difference, and β denote the output difference of m bits. If we denote the positions of the active 1-bit differences in the input and output as i and j, respectively, the input and output differences can be fixed with the following equations:

$$\overline{\alpha_0} \wedge \overline{\alpha_1} \wedge \cdots \wedge \alpha_i \wedge \cdots \wedge \overline{\alpha_{n-1}} \wedge \overline{\alpha_n} = 1,$$
$$\overline{\beta_0} \wedge \overline{\beta_1} \wedge \cdots \wedge \beta_j \wedge \cdots \wedge \overline{\beta_{m-1}} \wedge \overline{\beta_m} = 1.$$

From the perspective of diffusion property, our search considers Hamming weight of 1 to be better and other block cipher designers evaluated in the same way [24]–[26] when estimating the longest distinguisher. In addition, when constraints on active key differences at the input are imposed, evaluations are carried out while considering the condition of constraining the output differences to be 0. This is because there is a possibility that the difference between the key and the data path cancels out, resulting in no difference in the ciphertext. Given these constraints, for a block length of b and a key length of k bits, in a search space with a Hamming weight of 1 or less, we solve a total of $b^2 + k(b+1)$ SAT models in each round.

5. Evaluation Results

In this section, we show the search results of the longest distinguishers for the related-key impossible differential attack. Table 1 shows the summary of results with a comparison to existing results.

5.1 Evaluation Results of Each Target

The following shows search results of the related-key impossible differential distinguishers for Piccolo, TWINE and T-TWINE with the method described in Sect. 4.

Piccolo In the search for the distinguishers for Piccolo, we change the start round of the Piccolo-encryption and we find related-key impossible differential distinguishers up to 10 and 16 rounds of Piccolo-80 and Piccolo-128, respectively. Tables 4 and 5 show the length of related-key impossible differential distinguishers, the number of disitinguishers, and approximate time of the search under the conditions of a particular starting round for each key length of Piccolo.
According to Table 4 for Piccolo-80, we find related-key impossible differential distinguishers up to 10 rounds and they can be constructed regardless of the starting round for encryption. We also demonstrate that when encryption begins from the 3rd round, we can configure the longest 16-round distinguishers for Piccolo-128 in 16 different patterns.

TWINE Table 6 shows the number of related-key impossible differential disitinguishers and approximate time

[†]https://web.archive.org/web/20131022021257/http://www.sontrak.com/

Table 4 The number of related-key impossible differentials for Piccolo-80.

Start Round	Length of Distinguishers	Number of IDs	Approx.Time
1R	8R	592 IDs	2 hours 40 min.
	9R	32 IDs	2 hours 50 min.
	10R	32 IDs	3 hours
	11R	0 IDs	3 hours
2R	8R	64 IDs	2 hours 50 min.
	9R	64 IDs	2 hours 50 min.
	10R	32 IDs	3 hours
	11R	0 IDs	3 hours
3R	8R	16 IDs	2 hours 40 min.
	9R	320 IDs	2 hours 50 min.
	10R	32 IDs	3 hours
	11R	0 IDs	3 hours
4R	8R	1216 IDs	2 hours 40 min.
	9R	320 IDs	2 hours 50 min.
	10R	32 IDs	3 hours
	11R	0 IDs	3 hours

ID:impossible differential

Table 5 The number of related-key impossible differentials for Piccolo-128.

Start Round	Length of Distinguishers	Number of IDs	Approx.Time
1R	12R	176 IDs	4 hours 20 min.
	13R	32 IDs	4 hours 20 min.
	14R	16 IDs	5 hours
	15R	0 IDs	5 hours 10 min.
2R	11R	320 IDs	4 hours
	12R	48 IDs	4 hours 20 min.
	13R	16 IDs	4 hours 20 min.
	14R	0 IDs	5 hours
3R	14R	16 IDs	5 hours
	15R	16 IDs	5 hours 10 min.
	16R	16 IDs	4 hours 50 min.
	17R	0 IDs	5 hours 20 min.
4R	13R	16 IDs	4 hours 30 min.
	14R	16 IDs	5 hours 10 min.
	15R	16 IDs	5 hours 10 min.
	16R	0 IDs	4 hours 50 min.

ID:impossible differential

Table 6 The number of related-key impossible differentials for TWINE.

Target	length of Distinguishers	Number of IDs	Approx.Time
TWINE-80	13R	4102 IDs	1 hour 40 min.
	14R	1310 IDs	2 hours 20 min.
	15R	44 IDs	2 hours 40 min.
	16R	0 IDs	2 hours 40 min.
TWINE-128	16R	400 IDs	3 hours 40 min.
	17R	20 IDs	3 hours 50 min
	18R	8 IDs	4 hours
	19R	4 IDs	4 hours
	20R	0 IDs	4 hours

ID:impossible differential

of the search for TWINE-80 and TWINE-128 at each round. As for the searches of TWINE, we find related-key impossible differential distinguishers up to 15 and 19 rounds for 80-bit and 128-bit keys, respectively. For TWINE-80, we can configure 44 different disitnguishers for 15 rounds and for TWINE-128, we can configure 4 different disitinguishers for 19 rounds.

T-TWINE Table 7 shows the number of related-tweak and related-tweak-key impossible differential disitinguishers and approximate time of the search for T-TWINE-80 and T-TWINE-128 at each round. Under the related-tweak conditions, we find the distinguishers up to 18 rounds T-TWINE-80 and T-TWINE-128, respectively. Under the related-tweak-key conditions, we find the distinguishers up to 18 and 19 rounds 80-bit and 128-bit keys, respectively. We demonstrate that we can configure 4 different distinguishers for each key length of T-TWINE.

5.2 More Rounds for Piccolo-128

In this section, we present the 16-round impossible differentials discovered by our SAT-aided evaluation and manually extend it to the 17-round differentials by taking the cancelation of difference in the key and data processing part into account. These 17-round impossible differentials, where differences in plaintext and key are active, could not be discovered through the search under the condition of Hamming weight of 1 by the SAT-aided evaluation.

First, we denote the notations used in this section. $\Delta X, \Delta Y, \Delta K$ are the differences of plaintext, ciphertext and 128-bit master key, respectively. Δk_i and Δy_i mean the ith 16-bit of master key and ciphertext. From the above notation, the following can be obvious.
$\Delta K_{(128)} = \Delta k_0 | \Delta k_1 | \cdots | \Delta k_7$, $\Delta Y_{(64)} = \Delta y_0 | \Delta y_1 | \Delta y_2 | \Delta y_3$.
k_4^i denotes the ith bit of difference in k_4 is active, and otherwise inactive. Therefore, Hamming weight of k_4^i which is 16-bit length is 1.
$(\Delta X, \Delta K) \overset{16R}{\nrightarrow} (\Delta Y)$ means ΔX and ΔK never lead to ΔY in 16-round encryptions, which represents a 16-round impossible differential.
In this research, we demonstrated the following 16 kinds of impossible differentials.

$$(\Delta X = 0_{(64)}, 0_{(64)} | \Delta k_4^i | 0_{(48)}) \overset{16R}{\nrightarrow} (0_{(32)} | \Delta y_2^i | 0_{(16)}) \ (0 \le i \le 15).$$

We illustrate these impossible differentials in Fig. 3. In this figure, the gray area indicates that one bit within the area is active. When considering the relationship between round keys and the cancellation of plaintext differences, we can decrypto one more round with probability 1. Thus, we can extend the related-key impossible differentials of Piccolo-128 to 17 rounds. These differentials are expressed as follows and depicted in Fig. 4.

$$(0_{(32)} | \Delta x_2^i | 0_{(16)}, 0_{(64)} | \Delta k_4^i | 0_{(48)}) \overset{17R}{\nrightarrow} (0_{(32)} | \Delta y_2^i | 0_{(16)}) \ (0 \le i \le 15).$$

6. Consideration

In this section, we discuss the key recovery attack on Piccolo-128 with updated impossible differentials.

Table 7 The number of related-tweak and related-tweak-key impossible differentials for T-TWINE.

Target	Tweak/Tweak-Key	length of Distinguishers	Number of IDs	Approx.Time
T-TWINE-80	T	16R	76 IDs	2 hours 30 min.
		17R	16 IDs	2 hours 30 min.
		18R	4 IDs	2 hours 30 min.
		19R	0 IDs	2 hours 30 min.
	TK	16R	76 IDs	4 hours
		17R	16 IDs	4 hours 20 min.
		18R	4 IDs	4 hours
		19R	0 IDs	4 hours 30 min.
T-TWINE-128	T	16R	76 IDs	2 hours 30 min.
		17R	16 IDs	2 hours 30 min.
		18R	4 IDs	2 hours 30 min.
		19R	0 IDs	2 hours 30 min.
	TK	17R	36 IDs	5 hours 20 min.
		18R	12 IDs	5 hours 30 min.
		19R	4 IDs	5 hours 50 min.
		20R	0 IDs	5 hours 50 min.

ID:impossible differential
T:Tweak,TK:Tweak-Key

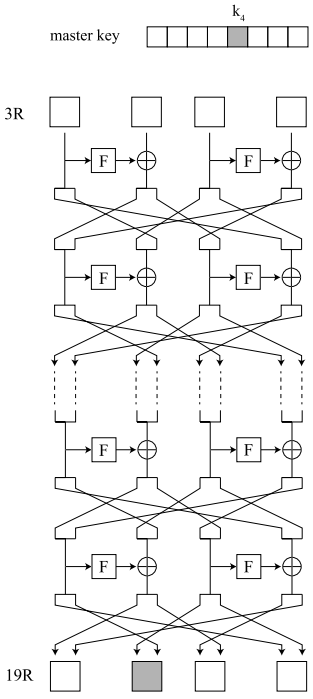

Fig. 3 16 rounds impossible differential of Piccolo-128.

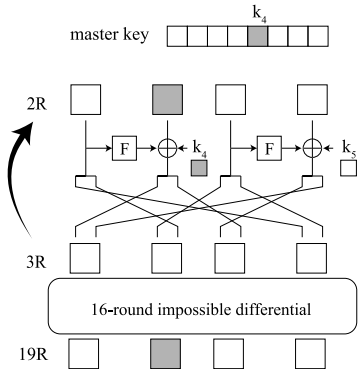

Fig. 4 17 rounds impossible differential of Piccolo-128.

The general approach in impossible differential attacks is to extend the impossible differential by some rounds at the top and the bottom of the impossible differential. Then, guess the key bits that intervene in these rounds and check whether a trial pair is partially encrypted (or decrypted) to the impossible differential. When the input-output differences extended to additional rounds reach the impossible differential, we know that the guessed key bits are certainly wrong and we can remove the key from the candidate key space.

When we conduct key recovery attacks using the impossible differential we discovered in Piccolo-128, because we could only consider intermediate rounds of impossible differentials with Hamming weight of 1, the probability of reaching them from the extended rounds was low, and we were unable to eliminate a sufficient number of key candidates. Therefore, it was impossible to conduct key recovery attacks utilizing the impossible differentials discovered under this condition. However, finding the longest distinguishers is crucial for examining the structural properties of the cipher.

7. Conclusion

In this paper, we performed bit-level evaluations on Piccolo, TWINE and T-TWINE against related-key impossible differential attacks by leveraging SAT-aided approaches. Especially, by changing the starting round of encryption of Piccolo, we succeeded in identifying the longest 16-round distinguisher on Piccolo-128 for the fisrt time. In addition, considering the cancellation of round key and plaintext difference, we demonstrated we can extend Piccolo-128 distinguishers found by the SAT approach to 17 rounds. For the evaluation of TWINE-128, we searched for related-key impossible differential distinguisher for the first time and we revealed the distinguishers up to 19 rounds. Regarding the

evaluation of T-TWINE under the related-tweak-key condition, we searched for the longest distinguishers for the first time and we revealed the distinguishers up to 18 rounds for 80-bit key and up to 19 rounds 128-bit key.

We explored impossible differentials under the condition that the Hamming weight of the active differences in the input and output is 1, respectively. However, it is likely that this condition is not effective in key recovery attacks. Therefore, it is necessary to consider methods for identifying impossible differentials that are effective against key recovery attacks in the future.

Acknowledgments

Takanori Isobe is supported by JST, PRESTO Grant Number JPMJPR2031. This research was in part conducted under a contract of "Research and development on new generation cryptography for secure wireless communication services" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan. This result is also obtained from the commissioned research (JPJ012368C05801) by National Institute of Information and Communications Technology (NICT), Japan. The authors would like to express their gratitude to Ryoma Ito for useful discussions.

References

[1] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An ultra-lightweight blockcipher," CHES, Lecture Notes in Computer Science, vol.6917, pp.342–357, Springer, 2011.

[2] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE: A lightweight block cipher for multiple platforms," Selected Areas in Cryptography, Lecture Notes in Computer Science, vol.7707, pp.339–354, Springer, 2012.

[3] S.A. Azimi, Z. Ahmadian, J. Mohajeri, and M.R. Aref, "Impossible differential cryptanalysis of piccolo lightweight block cipher," ISCISC, pp.89–94, IEEE, 2014.

[4] M. Minier, "On the security of piccolo lightweight block cipher against related-key impossible differentials," INDOCRYPT, Lecture Notes in Computer Science, vol.8250, pp.308–318, Springer, 2013.

[5] X. Zheng and K. Jia, "Impossible differential attack on reduced-round TWINE," ICISC, Lecture Notes in Computer Science, vol.8565, pp.123–143, Springer, 2013.

[6] A. Biryukov, P. Derbez, and L. Perrin, "Differential analysis and meet-in-the-middle attack against round-reduced TWINE," FSE, Lecture Notes in Computer Science, vol.9054, pp.3–27, Springer, 2015.

[7] Y. Wei, P. Xu, and Y. Rong, "Related-key impossible differential cryptanalysis on lightweight cipher TWINE," J. Ambient Intell. Humaniz. Comput., vol.10, no.2, pp.509–517, 2019.

[8] E. Biham, A. Biryukov, and A. Shamir, "Miss in the middle attacks on IDEA and Khufu," FSE, Lecture Notes in Computer Science, vol.1636, pp.124–138, Springer, 1999.

[9] J. Kim, S. Hong, J. Sung, C. Lee, and S. Lee, "Impossible differential cryptanalysis for block cipher structures," INDOCRYPT, Lecture Notes in Computer Science, vol.2904, pp.82–96, Springer, 2003.

[10] Y. Luo, X. Lai, Z. Wu, and G. Gong, "A unified method for finding impossible differentials of block cipher structures," Inf. Sci., vol.263, pp.211–220, 2014.

[11] L. Sun, D. Gérault, W. Wang, and M. Wang, "On the usage of deterministic (related-key) truncated differentials and multidimensional linear approximations for SPN ciphers," IACR Trans. Symmetric Cryptol., vol.2020, no.3, pp.262–287, 2020.

[12] Y. Sasaki and Y. Todo, "New impossible differential search tool from design and cryptanalysis aspects: Revealing structural properties of several ciphers," EUROCRYPT (3), Lecture Notes in Computer Science, vol.10212, pp.185–215, 2017.

[13] L. Sun, W. Wang, and M. Wang, "Accelerating the search of differential and linear characteristics with the SAT method," IACR Trans. Symmetric Cryptol., vol.2021, no.1, pp.269–315, 2021.

[14] M. Tolba, M. ElSheikh, and A.M. Youssef, "Impossible differential cryptanalysis of reduced-round tweakable TWINE," AFRICACRYPT, Lecture Notes in Computer Science, vol.12174, pp.91–113, Springer, 2020.

[15] G. Jakimoski and Y. Desmedt, "Related-key differential cryptanalysis of 192-bit key AES variants," Selected Areas in Cryptography, Lecture Notes in Computer Science, vol.3006, pp.208–221, Springer, 2003.

[16] L. Knudsen, "Deal-a 128-bit block cipher," Complexity, vol.258, no.2, 1998.

[17] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," EUROCRYPT, Lecture Notes in Computer Science, vol.1592, pp.12–23, Springer, 1999.

[18] E. Biham, "New types of cryptanalytic attacks using related keys," J. Cryptology, vol.7, no.4, pp.229–246, 1994.

[19] N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," Inscrypt, Lecture Notes in Computer Science, vol.7537, pp.57–76, Springer, 2011.

[20] F. Liu, G. Wang, S. Sarkar, R. Anand, W. Meier, Y. Li, and T. Isobe, "Analysis of RIPEMD-160: New collision attacks and finding characteristics with MILP," EUROCRYPT (4), Lecture Notes in Computer Science, vol.14007, pp.189–219, Springer, 2023.

[21] Y. Liu, Z. Xiang, S. Chen, S. Zhang, and X. Zeng, "A novel automatic technique based on MILP to search for impossible differentials," ACNS (1), Lecture Notes in Computer Science, vol.13905, pp.119–148, Springer, 2023.

[22] J. Erlacher, F. Mendel, and M. Eichlseder, "Bounds for the security of ascon against differential and linear cryptanalysis," IACR Trans. Symmetric Cryptol., vol.2022, no.1, pp.64–87, 2022.

[23] L. Sun, W. Wang, and M. Wang, "Linear cryptanalyses of three aeads with GIFT-128 as underlying primitives," IACR Trans. Symmetric Cryptol., vol.2021, no.2, pp.199–221, 2021.

[24] S. Banik, Z. Bao, T. Isobe, H. Kubo, F. Liu, K. Minematsu, K. Sakamoto, N. Shibata, and M. Shigeri, "WARP: Revisiting GFN for lightweight 128-bit block cipher," SAC, Lecture Notes in Computer Science, vol.12804, pp.535–564, Springer, 2020.

[25] S. Banik, T. Isobe, F. Liu, K. Minematsu, and K. Sakamoto, "Orthros: A low-latency PRF," IACR Trans. Symmetric Cryptol., vol.2021, no.1, pp.37–77, 2021.

[26] T. Isobe, R. Ito, F. Liu, K. Minematsu, M. Nakahashi, K. Sakamoto, and R. Shiba, "Areion: Highly-efficient permutations and its applications to hash functions for short input," IACR Trans. Cryptogr. Hardw. Embed. Syst., vol.2023, no.2, pp.115–154, 2023.

Shion Utsumi received the B.E. degree from University of Hyogo, Japan in 2022. He is currently a M.E. student at University of Hyogo, Japan. His research interest is cryptography.

Kosei Sakamoto received the B.E., M.E., and Ph.D. degrees from Kansai University, Japan, in 2017, and University of Hyogo, Japan, in 2020 and 2023, respectively. He has worked at Mitsubishi Electric Corporation from 2023. His current research interests include information security and cryptography.

Takanori Isobe received the B.E., M.E., and Ph.D. degrees from Kobe University, Japan, in 2006, 2008 and 2013, respectively. From 2008 to 2017, he worked at the Sony Corporation. From 2017 to 2022, he has been an Associate Professor at University of Hyogo. Since 2023, he has been a Professor at University of Hyogo.