PAPER

Improving the Security Bounds against Differential Attacks for Pholkos Family

Nobuyuki TAKEUCHI[†a)], ***Nonmember***, **Kosei SAKAMOTO**[††], ***Member***, **Takuro SHIRAYA**[†], ***Nonmember***, ***and*** **Takanori ISOBE**[†,†††], ***Member***

SUMMARY At CT-RSA 2022, Bossert et al. proposed Pholkos family, an efficient large-state tweakable block cipher. In order to evaluate the security of differential attacks on Pholkos, they obtained the lower bounds for the number of active S-boxes for Pholkos using MILP (Mixed Integer Linear Programming) tools. Based on it, they claimed that Pholkos family is secure against differential attacks. However, they only gave rough security bounds in both of related-tweak and related-tweakey settings. To be more precise, they estimated the lower bounds of the number of active S-boxes for relatively-large number of steps by just summing those in the small number of steps. In this paper, we utilize efficient search methods based on MILP to obtain tighter lower bounds for the number of active S-boxes in a larger number of steps. For the first time, we derive the exact minimum number of active S-boxes of each variant up to the steps where the security against differential attacks can be ensured in related-tweak and related-tweakey settings. Our results indicate that Pholkos-256-128/256-256/512/1024 is secure after 4/5/3/4 steps in the related-tweak setting, and after 5/6/3/4 steps in the related-tweakey setting, respectively. Our results enable reducing the required number of steps to be secure against differential attacks of Pholkos-256-256 in related-tweak setting, and Pholkos-256-128/256 and Pholkos-1024 in the related-tweakey setting by one step, respectively.

key words: Pholkos*, differential attack, active S-box, MILP*

1. Introduction

1.1 Background

The differential attack, first introduced by Biham and Shamir [1], is widely regarded as one of the most important cryptanalytic techniques for symmetric-key primitives [2]–[4]. New designs for symmetric-key primitives must be able to be secure against this type of attack as it is considered a crucial security requirement. However, evaluating the security against differential attacks can be computationally demanding, as the designer needs to exhaustively evaluate the space that an attacker can exploit. To address this issue, in many cases the designers take the aid of mathematical solvers, such as an MILP (Mixed Integer Linear Programming) [5]–[8] and SAT (Boolean Satisfiability Problem) [9]–[12].

At CT-RSA 2022, Bossert et al. proposed an efficient large-state tweakable block cipher called Pholkos [13]. With support for 256-bit keys, Pholkos offers 128-bit security against quantum key-recovery attacks. Moreover, even if Pholkos is used in modes having birthday security the scheme has practically sufficient security. In order to evaluate the security of differential attacks on Pholkos, designers obtained the lower bounds for the number of active S-boxes using MILP tools. Based on it, they claimed that Pholkos family is secure against differential attacks. However, they only gave rough security bounds in both of related-tweak and related-tweakey settings due to the computational hardness of the security evaluation that comes from the large internal state. To be more precise, they estimated the lower bounds of the number of active S-boxes for relatively-large number of steps by just summing those in the small number of steps.

1.2 Our Contribution

In this paper, we utilize efficient search methods based on MILP to obtain lower bounds for the number of active S-boxes in a larger number of steps. For the first time, we derive the exact minimum number of active S-boxes of each variant up to the steps where the security against differential attacks can be ensured in related-tweak and related-tweakey settings. Our results indicate that Pholkos-256-128/256-256/512/1024 is secure after 4/5/3/4 steps in the related-tweak setting, and after 5/6/3/4 steps in the related-tweakey setting, respectively. Our results enable reducing the required number of steps to be secure against differential attacks of Pholkos-256-256 in related-tweak setting, and Pholkos-256-128/256 and Pholkos-1024 in the related-tweakey setting by one step, respectively. A summary of our results is shown in Table 1. The details of our contributions in each setting are summarized as follows:

(1) Single-key setting

Although the required number of rounds for differential attacks is not changed from the designer's evaluation, we show the exact lower bounds for the number of active S-boxes over all rounds for Pholkos-512 for the first time and those for more steps for Pholkos-1024 than the designer's evaluation [13].

(2) Related-tweak setting

We reveal the lower bounds for the number of active S-boxes over all rounds for Pholkos-256-128/256-256 in the related-tweak setting. We show that Pholkos-256-128 can be secure

Manuscript received August 23, 2023.
Manuscript revised November 22, 2023.
Manuscript publicized February 8, 2024.
[†]Graduate School of Information Science, University of Hyogo, Kobe-shi, 650-0047 Japan.
[††]Mitsubishi Electric Corporation, Tokyo, 100-8310 Japan.
[†††]National Institute of Information and Communications Technology, Tokyo, 184-8795 Japan.
a) E-mail: nobuyuki.takeuchi0722@gmail.com
DOI: 10.1587/transfun.2023EAP1098

Table 1 Summary of the required number of steps to be secure against differential attacks for Pholkos family in each setting.

Setting	Variant	[13]	Our
Single-key	Pholkos-256-128	4/8	4/8
	Pholkos-256-256	4/8	4/8
	Pholkos-512	3/10	3/10
	Pholkos-1024	4/12	4/12
Related-tweak	Pholkos-256-128	5/8	**4/8**
	Pholkos-256-256	5/8	5/8
	Pholkos-512	3/10	3/10
	Pholkos-1024	4/12	4/12
Related-tweakey	Pholkos-256-128	6/8	**5/8**
	Pholkos-256-256	7/8	**6/8**
	Pholkos-512	3/10	3/10
	Pholkos-1024	5/12	**4/12**

against differential attacks after 4 steps, which improves the designer's result by one round. For Pholkos-512 we derive the bounds up to 7 steps while the designer's evaluation [13] can obtain it only up to 4 steps. For Pholkos-1024, we update the number of lower bounds of active S-boxes in 5 steps.

(3) Related-tweakey setting

We update the lower bounds for the number of active S-boxes in 3 steps of Pholkos-256-256. Moreover, we first derived the lower bounds for the number of active S-boxes over all rounds for Pholkos-256-128 in the related-tweakey setting. Our results show that for Pholkos-256-128/256-256/1024, 5/6/4 steps are sufficient to be secure against differential attacks, respectively, which update the designer's result of 6/7/5 steps, respectively.

1.3 Organization

This paper is organized as follows. Section 2 provides the differential attack and how to evaluate the security against this attack with an MILP. Section 3 describes the specification of Pholkos-256-128/256-256/512/1024. In Sect. 4, we explain our method to evaluate the security of Pholkos in the single-key, related-tweak, and related-tweakey settings. In Sect. 5, we show our evaluation results and Sect. 6 discusses them. Finally, we conclude this paper in Sect. 7.

2. Preliminaries

This section describes differential attacks and the method to evaluate active S-boxes with MILP.

2.1 Differential Attack

The differential attack [1] is one of the most popular cryptanalysis techniques for block ciphers. In order to evaluate the resistance against differential attacks, we evaluate its differential probability DP_{f_b} for all possible differences and obtain the maximum one called maximum differential probability $DP_{f_b max}$. Let f_b, Δx_0, and Δx_r be the b-bit block cipher, differences of plaintexts, and differences of ciphertexts, respectively. DP_{f_b} is defined as follows:

$$DP_{f_b}(\Delta x_0, \Delta x_r) = \frac{\#\{x \in \{0,1\}^b | f_b(x) \oplus f_b(x \oplus \Delta x_0) = \Delta x_r\}}{2^b}.$$

If b is small, calculating $DP_{f_b max}$ is feasible. However, this is not the case for ciphers having more than a 64-bit block due to computational hardness. Therefore, the maximum differential characteristic probability $DCP_{f_b max}$ is utilized to approximate $DP_{f_b max}$. $DCP_{f_b max}$ is defined as a product of the differential characteristic probability DCP_{f_b} for each round as follows:

$$DCP_{f_b} = \prod_{R=1}^{r} DP_{f_b}(\Delta x_R, \Delta x_{R+1}),$$

$$DCP_{f_b max} = \max_{\substack{\Delta x_1 \neq 0 \\ \Delta x_2, \ldots, \Delta x_{r+1}}} DCP_{f_b},$$

where r is the number of rounds. For block ciphers whose non-linear layer consists of only an S-box, $DCP_{f_b max}$ can be calculated by multiplication of DP_{smax} for all active S-boxes under a well-known Markov cipher assumption, i.e. $DCP_{f_b max} = (DP_{smax})^{AS_{lbD}}$ where DP_{smax} and AS_{lbD} are the maximum differential probability of the S-box and the lower bound for the number of active S-boxes, respectively.

2.2 MILP-Aided Security Evaluation

MILP (Mixed Integer Linear Programming) is one of the methods for efficiently finding variables to maximize or minimize a particular objective function under some constraints expressed by a linear inequality. Mouha et al. introduced MILP into the field of symmetric-key cryptography to efficiently evaluate the lower bound for the number of active S-boxes [5]. In their evaluation, we need to express all operations in a symmetric-key primitive as linear inequalities and assign them to the MILP model as constraints. Then, the total number of active S-boxes is assigned to the MILP model as the objective function. After constructing the MILP model, we can obtain the lower bound for the number of active S-boxes by giving it to an MILP solver. In this study, we employ the Gurobi Optimizer [14] as the MILP solver.

3. Specification of Pholkos

The family of Pholkos was proposed by Bossert et al. at CT-RSA 2022 [13]. Algorithm 1 shows the encryption procedures of Pholkos-n-t. Pholkos is specified as four variants by the length of the plaintext and the tweak. Table 2 shows each variant and its security level (claimed security) in the single-key, related-tweak, and related-tweakey settings.

3.1 Notations

- r – the number of rounds.
- s – the number of steps. One step proceeds two rounds

of AES round function and 2 tweakey XORs, i.e. $s = r/2$.

- P – the plaintext of Pholkos. The plaintext is divided into 128-bit substates, i.e. $P = P[0], P[1], \ldots, P[(n/128) - 1]$. It is also divided into eight 32-bit words. i.e. $P = (P_0, P_1, \ldots, P_{(n/32)-1})$.
- IK – 256-bit initial key. It is expressed as follows:

$$IK = IK_0, IK_1, \ldots, IK_7.$$

- K^i – n-bit i-th round expanded key ($0 \leq i \leq r$). It is expressed as follows:

$$K^i = K^i_0, K^i_1, \ldots, K^i_{(n/32)-1},$$
$$K^i = K^i[0], K^i[1], \ldots, K^i[(n/128) - 1].$$

- T^i – t-bit i-th round tweak ($0 \leq i \leq r$). It is expressed as follows:

$$T^i = T^i_0, T^i_1, \ldots, T^i_{(n/32)-1},$$
$$T^i = T^i[0], T^i[1], \ldots, T^i[(t/128) - 1].$$

- RTK^i – n-bit i-th round tweakey ($0 \leq i \leq r$). It is expressed as follows:

$$RTK^i = RTK^i[0], RTK^i[1], \ldots, RTK^i[(n/128) - 1].$$

- RC^i – The 128-bit i-th round constant ($0 \leq i \leq r$), which is defined as in [13].
- π_n – The 32-bit-wise permutation shown in Table 3, where n is the length of input. For $n = 1024$, there are two type permutations $\pi_{1024,0}$ and $\pi_{1024,1}$, which are applied alternatingly, starting with $\pi_{1024,0}$.
- π_τ – The byte-wise permutation shown in Table 4. π_τ is used to update subkeys and tweaks.
- $A(X)$ – one AES round function applied to the substate X, is defined as follows:

$$A(X) = \text{MixColumns} \circ \text{ShiftRows} \circ \text{SubBytes}(X),$$

where MixColumns, ShiftRows, and SubBytes are the same operations as in AES.

- $A_{last}(X)$ – last AES round function applied to the substate X, as defined follows:

$$A_{last}(X) = \text{ShiftRows} \circ \text{SubBytes}(X).$$

3.2 Step Function

Figure 1 illustrates the first step of Pholkos-512 as an example of the step function of Pholkos family. At the beginning of the first step, the plaintext is first XORed to the pre-whitening RTK^0. Then, it will proceed into the step function. One step consists of two rounds and the application of π_n. Each round consists of $A(X)$ and XORed RTK^i. The final step excludes the final π_n and employs $A_{last}(X)$ instead of $A(X)$.

3.3 Tweakey Scheduling

We show the overview of the tweakey scheduling function in Fig. 2. The detailed procedure is described as follows.

(1) Key Expansion

For variants with $n > 256$, the key expansion φ shown in Fig. 3 is first applied to the initial key. Function φ expand K

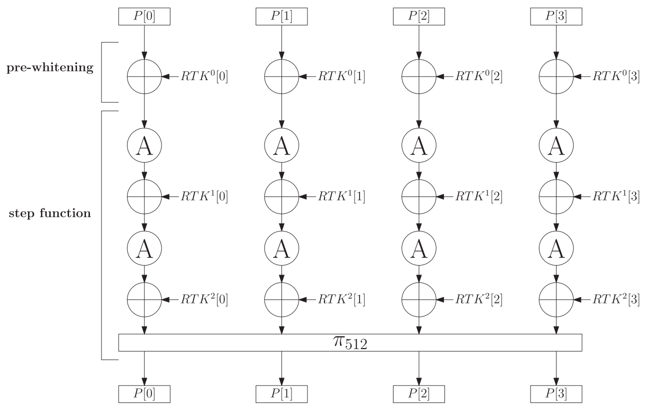


Fig. 1 First step of Pholkos-512 including pre-whitening RTK^0. The A denotes $A(X)$.

Table 2 Parameters of Pholkos. n and t denote the size of the plaintext and tweak, respectively.

Variant	Length (bits)			#steps	Security (bits)		
	Plaintext (n)	Key	Tweak (t)		SK	RT	RTK
Pholkos-256-128	256	256	128	8	256	256	256
Pholkos-256-256	256	256	256	8	256	256	256
Pholkos-512	512	256	128	10	256	256	256
Pholkos-1024	1024	256	128	12	256	256	256

SK : Single-key setting
RT : Related-tweak setting
RTK : Related-tweakey setting

Table 3 32-bit-wise permutation π_n, where n is input length.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\pi_{256}(i)$	0	5	2	7	4	1	6	3																								
$\pi_{512}(i)$	0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11																
$\pi_{1024,0}(i)$	0	9	18	27	4	13	22	31	8	17	26	3	12	21	30	7	16	25	2	11	20	29	6	15	24	1	10	19	28	5	14	23
$\pi_{1024,1}(i)$	0	5	2	7	4	9	6	11	8	13	10	15	12	17	14	19	16	21	18	23	20	25	22	27	24	29	26	31	28	1	30	3

Table 4 Byte-wise permutation π_τ.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi_\tau(i)$	11	12	1	2	15	0	5	6	3	4	9	10	7	8	13	14

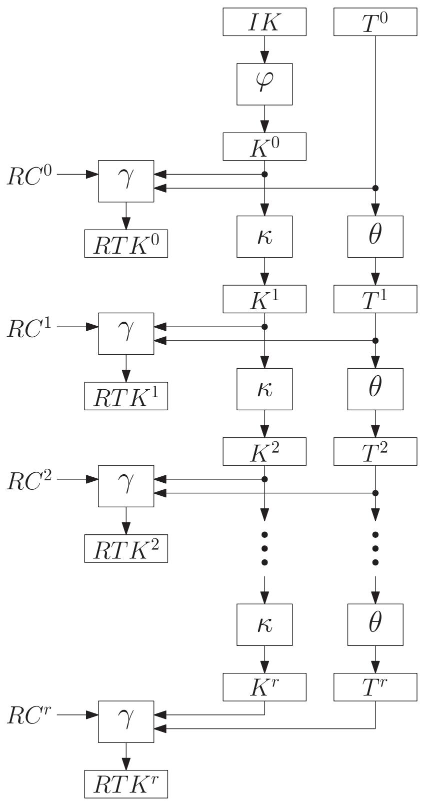


Fig. 2 Tweakey schedule of Pholkos.

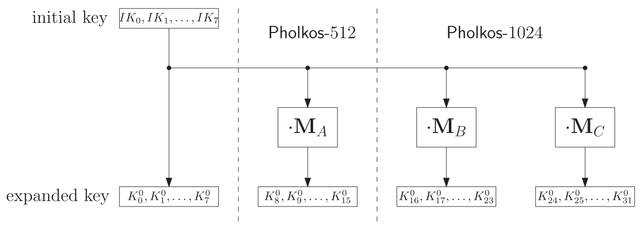


Fig. 3 Key expansion function φ.

using IK and binary circulant matrices $\mathbf{M}_A$, $\mathbf{M}_B$, and $\mathbf{M}_C \in (\mathbb{F}_{2^{32}})^{8\times 8}$ with branch numbers of four:

$$\mathbf{M}_A = \mathsf{circ}(11001000), \qquad \mathbf{M}_B = \mathsf{circ}(10101000),$$
$$\mathbf{M}_C = \mathsf{circ}(10011000).$$

For Pholkos-512, the expanded key $\mathbf{K}$ meets the following conditions:

$$\mathbf{K} = \mathbf{IK} \parallel \mathbf{M}_A \cdot \mathbf{IK}^\top.$$

For Pholkos-1024, the expanded key $\mathbf{K}$ meets the following conditions:

$$\mathbf{K} = \mathbf{IK} \parallel \mathbf{M}_A \cdot \mathbf{IK}^\top \parallel \mathbf{M}_B \cdot \mathbf{IK}^\top \parallel \mathbf{M}_C \cdot \mathbf{IK}^\top.$$

(2) Generation of Tweakey

In order to generate the tweakey schedule, we need to obtain the tweak and key schedule from the initial tweak T^0 and (expanded) key K^0. The function θ and κ provide T^i and K^i from T^{i-1} and K^{i-1} in Figs. 4(a) and 4(b), respectively. Subsequently, RTK^i is generated using the function γ that combines T^i, K^i and RC^i as shown in Fig. 4(c). Each function is performed as below:

θ **(Fig. 4(a))** : The function θ obtains T^i from T^{i-1}, i.e. $T^i = \theta(T^{i-1})$. If $t = 128$, θ processes π_τ. When $t = 256$, θ first executes T^{i-1} using π_{256} and generates intermediate tweak $T^{i-1\prime}$. Next, $T^{i-1\prime}$ is split into 128-bit substates, and π_τ is processed on each substate.

κ **(Fig. 4(b))** : The function κ obtains K^I from K_{I-1}, I.e, $K^i = \kappa(K^{i-1})$. For the variant has an n-bit plaintext, π_n is first applied to K^{I-1}, and then the output of π_n is divided into 128-bit states, I.e, $\pi_n(K^{I-1}) = (K^{I-1\prime}[0]||K^{I-1\prime}[1]||\ldots||K^{I-1\prime}[(n/128) - 1])$ where $K^{I-1\prime}[j] \in GF(2)$. Lastly, π_τ is applied to each $K^{I-1\prime}[j]$.

γ **(Fig. 4(c))** : The function γ obtains RTK^i from T^i, K^i, and RC^i, i.e. $RTK^i = \gamma(T^i, K^i, RC^i)$. If $t = 128$, let j be the index of substate, γ has been computed as follows:

$$RTK^i[j] = \begin{cases} T^i[0] \oplus K^i[j] \oplus RC^i & \text{if } j = 0, \\ T^i[0] \oplus K^i[j] & \text{otherwise.} \end{cases}$$

When $t = 256$, γ has been computed as follows:

$$RTK^i[j] = \begin{cases} T^i[j] \oplus K^i[j] \oplus RC^i & \text{if } j = 0, \\ T^i[j] \oplus K^i[j] & \text{otherwise.} \end{cases}$$

These functions are iterated until generating RTK^r.

4. Our MILP Model for Security Evaluation

In this section, we describe how to evaluate the lower bound for the number of active S-box in Pholkos. We evaluate it based on a byte-wise truncated difference, as Pholkos is constructed by the byte-wise operations. To guarantee 256-bit security against the distinguishing attack, more than 42 active S-boxes are needed, as the maximum differential probability of an S-box in AES is 2^{-6}. In order to assign the byte-wise MILP model, we first need to define byte-wise truncated difference vector $\Delta\mathbf{x} = (\Delta x_0, \Delta x_1, \ldots, \Delta x_7)$ as follows:

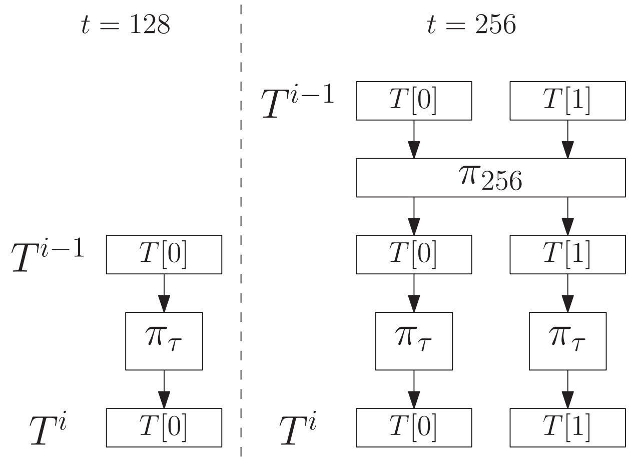
$$\Delta x = \begin{cases} 0 & \text{if } \displaystyle\sum_{i=0}^{7} \Delta x_i = 0, \\ 1 & \text{otherwise.} \end{cases}$$

Based on this, we give the constraints to express the differential propagation of each operation and the objective function according to [5]. In the following sections, we denote $\mathcal{M}_{var}$, $\mathcal{M}_{con}$, and $\mathcal{M}_{obj}$ as the set of variables, the set of constraints, and the objective function in an MILP model, respectively.

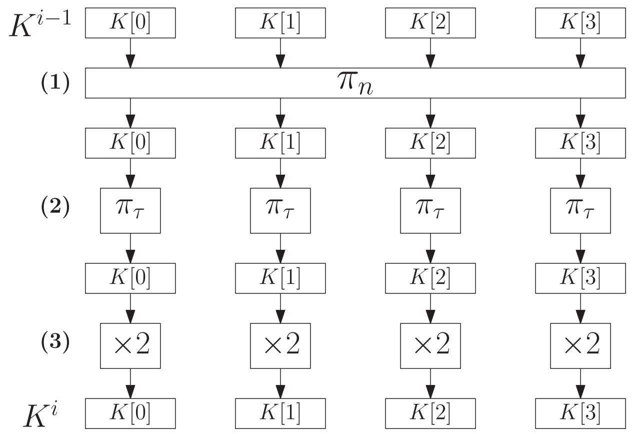
4.1 Basic MILP Model for Pholkos Family

(1) Constraints

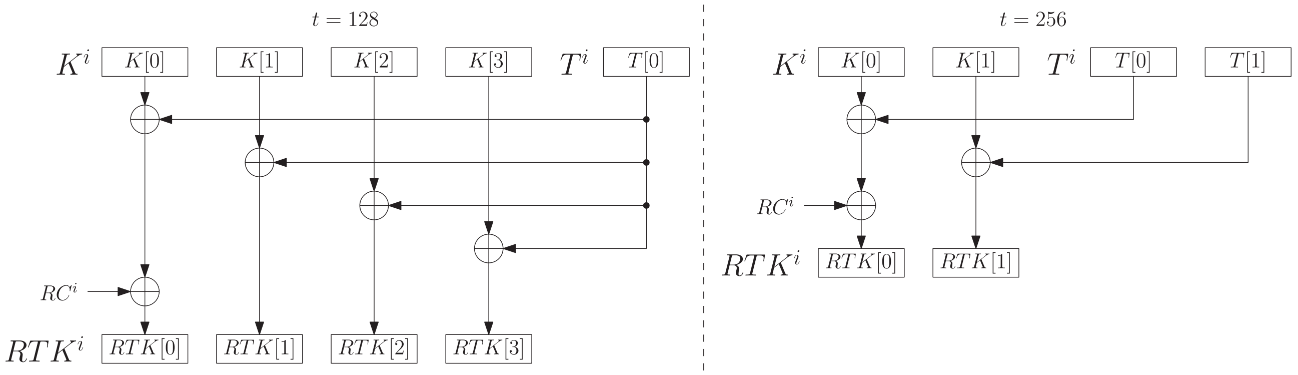
As Pholkos consists of an XOR, Linear transformation, and



(a) Function θ.



(b) Function κ.



(c) Function γ.

Fig. 4 Components of the tweakey generated functions.

S-box, we describe the model of the differential propagation.

XOR : Let input difference vectors be x_1, x_2. An output difference vector $y = x_1 \oplus x_2$ is expressed as follows:

$$\begin{cases} \mathcal{M}_{var} \leftarrow x_1, x_2, y \text{ as binary}, \\ \mathcal{M}_{con} \leftarrow -x_1 + x_2 + y \geq 0, \\ \mathcal{M}_{con} \leftarrow x_1 - x_2 + y \geq 0, \\ \mathcal{M}_{con} \leftarrow x_1 + x_2 - y \geq 0. \end{cases}$$

Remarks. In the evaluation of the truncated differences, we have to consider the case of $1 \oplus 1 = 1$ with the case of $1 \oplus 1 = 0$, i.e., $1 \oplus 1 = 0$ or 1.

Copy : Let an input difference vector be x. Output difference vectors $(y_1, y_2) = (x, x)$ are expressed as follows:

$$\begin{cases} \mathcal{M}_{var} \leftarrow x, y_1, y_2 \text{ as binary}, \\ \mathcal{M}_{con} \leftarrow x = y_1 = y_2. \end{cases}$$

Linear transformation : Let input difference vectors be $x_0, x_1, \ldots, x_n$, output difference vectors be $y_0, y_1, \ldots, y_n$, and the branch number be $\mathcal{B}$. The linear transformation is assigned as follows:

$$\begin{cases} \mathcal{M}_{var} \leftarrow x_0, x_1, \ldots, x_n, y_0, y_1, \ldots, y_n \text{ as binary}, \\ \mathcal{M}_{con} \leftarrow \sum_{i=0}^{n} x_i + \sum_{i=0}^{n} y_i \geq \mathcal{B} \cdot d, \\ \mathcal{M}_{con} \leftarrow d \geq x_i (0 \leq i \leq n), \\ \mathcal{M}_{con} \leftarrow d \geq y_i (0 \leq i \leq n), \end{cases}$$

where d is A binary dummy variables.

S-box : Let an input difference vector be x. An output difference vector $y = \text{SubBytes}(x)$ is expressed as follows:

$$\begin{cases} \mathcal{M}_{var} \leftarrow x, y \text{ as binary}, \\ \mathcal{M}_{con} \leftarrow x = y. \end{cases}$$

(2) Objective Function

The objective function is defined as the total number of active S-boxes. Let the S-box input difference vectors be $s_0, s_1, \ldots, s_n$, the objective function is expressed as follows:

$$\mathcal{M}_{obj} \leftarrow \text{Minimize}\left(\sum_{i=0}^{n} s_i\right).$$

If it is more than 42, the variant guarantees 256-bit security against the differential attack. We evaluate the resistance of Pholkos against differential attacks by assigning the above constraints in accordance with the encryption scheme of it

Algorithm 1 Specification of Pholkos-n-t.

```
procedure Encrypt($P, IK, T^0$)
    $RTK \leftarrow$ TweakeySchedule($IK, T^0$)
    $X^0 \leftarrow P \oplus RTK^0$
    for $i = 1$ to $s - 1$ do
        $Y^{2\cdot i} \leftarrow$ Step($X^{2\cdot(i-1)}$)
        $X^{2\cdot i} \leftarrow$ PermWords($i, Y^{2\cdot i}$)
    $Y^{2\cdot s} \leftarrow$ Step($X^{2\cdot(s-1)}$)
    return $Y^{2\cdot s}$
procedure Step($X^i$)
    $(X^i[0], X^i[1], \ldots, X^i[(n/128)-1]) \leftarrow X^i$
    for $j = 1$ to $2$ do
        for $k = 0$ to $(n/128) - 1$ do
            if $i + j < r$ then
                $X^{i+j}[k] \leftarrow \mathrm{A}(X^{i+j-1}[k]) \oplus RTK^{i+j}[k]$
            else
                $X^{i+j}[k] \leftarrow \mathrm{A}_{last}(X^{i+j-1}[k]) \oplus RTK^{i+j}[k]$
    return $X^{i+2}$
procedure PermWords($p, Y^i$)
    if $n = 1024$ then
        $\pi \leftarrow \pi_{1024,(p-1) \bmod 2}$
    else
        $\pi \leftarrow \pi_n$
    $(Y^i_0, Y^i_1, \ldots, Y^i_{(n/32)-1}) \leftarrow Y^i$
    for $j = 0$ to $(n/32) - 1$ do
        $X^i_j \leftarrow Y^i_{\pi(j)}$
    return $X^i$
procedure TweakeySchedule($IK, T^0$)
    $K^0 \leftarrow \varphi(IK)$
    $RTK^0 \leftarrow \gamma(RC^0, K^0, T^0)$
    for $i = 1$ to $r$ do
        $T^i \leftarrow \theta(T^{i-1})$
        $K^i \leftarrow \kappa(2, i-1, K^{i-1})$
        $RTK^i \leftarrow \gamma(RC^i, K^i, T^i)$
    return $(RTK^0, RTK^1, \ldots, RTK^r)$
procedure $\varphi(IK)$
    if $n = 256$ then return $IK$
    return $\mathsf{msb}_n(IK \,\|\, \mathbf{M}_A \cdot IK \,\|\, \mathbf{M}_B \cdot IK \,\|\, \mathbf{M}_C \cdot IK)$
procedure $\theta(T^{i-1})$
    if $t = 256$ then return $\kappa(1, i-1, K^{i-1})$
    return $\tau(T^{i-1})$
procedure $\kappa(\alpha, p, K^{i-1})$
    if $n = 1024$ then
        $\pi \leftarrow \pi_{1024,(p) \bmod 2}$
    else
        $\pi \leftarrow \pi_n$
    $(K^i_0, K^i_1, \ldots, K^i_{(n/32)-1}) \leftarrow K^i$
    for $j = 0$ to $(n/32) - 1$ do
        $K^i_j \leftarrow K^{i-1}_{\pi(j)}$
    $K^i \leftarrow \tau(K^i)$
    for $j = 0$ to $(n/128) - 1$ do
        $(K^i[j][0], K^i[j][1], \ldots, K^i[j][15]) \leftarrow K^i[j]$
        for $b = 0$ to $15$ do
            $K^i[j][b] \leftarrow \alpha \cdot K^i[j][b]$
    return $K^i$
procedure $\gamma(RC^i, K^i, T^i)$
    $RTK^i[0] \leftarrow T^i[0] \oplus K^i[0] \oplus RC^i$
    for $j = 1$ to $(n/128) - 1$ do
        if $t = 256$ then
            $RTK^i[j] \leftarrow T^i[j] \oplus K^i[j]$
        else
            $RTK^i[j] \leftarrow T^i[0] \oplus K^i[j]$
    return $RTK^i$
procedure $\tau(X)$
    for $i = 0$ to $(n/128) - 1$ do
        $(X[i][0], X[i][1], \ldots, X[i][15]) \leftarrow X[i]$
        for $b = 0$ to $15$ do
            $Z[i][b] \leftarrow X[i][\pi_\tau(b)]$
    return $Z$
```

in Algorithm 1.

In this paper, we evaluate it in the single-key, related-tweak, related-tweakey settings. The condition of these settings are as follows:

4.2 MILP Model for Each Attack Scenario

We need to give additional constraints to the input difference depending on the attack scenarios. In this paper, we evaluate the security against differential attacks in the single and related-tweak, and related tweakey settings. Hence, we show additional constraints about these settings one by one.

(1) Single-key Setting

The adversary can require a ciphertext corresponding to arbitrary plaintext to an encryption oracle. Therefore, the attacker can exploit arbitrary differences in a plaintext. We assign the setting to the constraint in MILP. Let input message differences be $x_0, x_1, \ldots, x_n$, we assign the constraint as follows:

$$\begin{cases} \mathcal{M}_{var} \leftarrow x_0, x_1, \ldots, x_n \text{ as binary}, \\ \mathcal{M}_{con} \leftarrow \sum_{a=0}^{n} x_a \geq 0. \end{cases}$$

(2) Related-tweak Setting

The adversary can require a ciphertext corresponding to arbitrary plaintext and tweak to an encryption oracle. Therefore, the attacker can exploit arbitrary differences in a plaintext and tweak. We assign the setting to the constraint in MILP. Let input message differences be $x_0, x_1, \ldots, x_m$ and input tweak differences be $t_0, t_1, \ldots, t_n$, we assign the constraint as follows:

$$\begin{cases} \mathcal{M}_{var} \leftarrow x_0, x_1, \ldots, x_m, t_0, t_1, \ldots, t_n \text{ as binary}, \\ \mathcal{M}_{con} \leftarrow \sum_{a=0}^{m} x_a + \sum_{b=0}^{n} t_b \geq 0. \end{cases}$$

(3) Related-tweakey Setting

The adversary can require a ciphertext corresponding to arbitrary plaintext, tweak, and key to an encryption oracle. Therefore, the attacker can exploit arbitrary differences in a plaintext, tweak, and key. We assign the setting to the constraint in MILP. Let input message differences be $x_0, x_1, \ldots, x_l$, input tweak differences be $t_0, t_1, \ldots, t_m$, and input key differences be $k_0, k_1, \ldots, k_n$, we assign the constraint as follows:

Algorithm 2 Example of dedicated constraints of Pholkos-256 in 2 steps in the related-tweakey setting.

```
1:  procedure RTKProcess($\mathcal{M}$)
2:      $\mathcal{M}, key, RTK \leftarrow$ Preprocess($\mathcal{M}$)
3:      $cnt[0] \leftarrow RTK[0][0] + RTK[1][5] + RTK[2][22] + RTK[3][7] + RTK[4][28]$
4:      $cnt[1] \leftarrow RTK[0][1] + RTK[1][2] + RTK[2][3] + RTK[3][8] + RTK[4][13]$
5:      ......
6:      $cnt[31] \leftarrow RTK[0][31] + RTK[1][4] + RTK[2][25] + RTK[3][26] + RTK[4][27]$
7:      for $i = 0$ to $31$ do
8:          $\mathcal{M}_{con} \leftarrow cnt[i] \geq 4 \cdot key[0][i]$
        return $\mathcal{M}$
9:  procedure Preprocess($\mathcal{M}$)
10:     for $r = 0$ to $4$ do
11:         for $i = 0$ to $31$ do
12:             $\mathcal{M}_{con} \leftarrow key[r][i]$ as binary
13:             $\mathcal{M}_{con} \leftarrow RTK[r][i]$ as binary
        return $\mathcal{M}, key, RTK$
```

$$\begin{cases} \mathcal{M}_{var} \leftarrow x_0, x_1, \dots, x_l, t_0, t_1, \dots, t_m, k_0, k_1, \dots, k_n \text{ as binary}, \\ \mathcal{M}_{con} \leftarrow \sum_{a=0}^{l} x_a + \sum_{b=0}^{m} t_b + \sum_{c=0}^{n} k_c \geq 0. \end{cases}$$

The byte-wise evaluation in this setting causes the problem that the number of active S-boxes is zero. If all bytes of the initial plaintext are inactive and all bytes of both the initial key and the initial tweak are active, all S-boxes are always inactive because all differences in the tweakey scheduling function are canceled. However, this never happens in the actual attack because a bit rotation in each word is applied in the function κ. Hence, we have to add constraints to avoid the case of all S-boxes being inactive. In this paper, as well [13], we apply the assumption that each active byte in RTKs are only canceled at most once. Based on this, we add constraints that the RTK through a one-byte active key path is canceled at most once. Specifically, focusing on the RTK generation process, if an input key is active, the total number of active RTKs by XORed operations of keys and tweaks is greater than or equivalent to the number of rounds multiplied by the input key minus one. We assign it to the MILP model as constraints. We show the example of Pholkos-256 in 2 steps in Algorithm 2.

5. Our Result by MILP-Based Search

In this section, we show the result for the lower bound on the number of active S-boxes for Pholkos family in each setting by MILP-based search. Our evaluations totally required three weeks with three computers equipped with AMD Ryzen Threadripper 3990X (64-Core) with 256 GB RAM.

Table 5 summarizes our results of lower bounds of active S-boxes for Pholkos.

Single-key Setting: Our results show that Pholkos-256-128/256-256/512/1024 required 4/4/3/4 steps to be secure against differential attacks in the single-key setting, respectively, i.e. after these steps, each variant has more than 42 active S-boxes.

Related-tweak Setting: We show that Pholkos-256-128/256-256/512/1024 require 4/5/3/4 steps, respectively, to be secure against differential attacks in the related-tweak setting. We derived the lower bounds for the number of active S-boxes of Pholkos-256-128/256-256/512/1024 up to 8/9/7/5 steps, respectively.

Related-tweakey Setting: We show that the required number of steps is estimated as 5/6/3/4 steps for Pholkos-256-128/256-256/512/1024 against differential attacks in the related-tweakey setting.

6. Updating Security Bounds

In this section, we explain how our new bounds of the number of active S-boxes affect the security of Pholkos. For the first time, we derive the exact minimum number of active S-boxes of each variant up to the steps where the security against differential attacks can be ensured in related-tweak and related-tweakey settings, while previous results estimated the lower bounds of the number of active S-boxes for relatively-large number of steps by just summing those in the small number of steps.

6.1 Single-Key Setting

While existing results [13] presented the exact lower bounds for the number of active S-boxes for 8 steps for Pholkos-256-128/256-256/512 and those for 7 steps for Pholkos-1024, respectively, we show the lower bounds of Pholkos-256-128/256-256/512/1024 up to 10 steps for the first time. Although our results cannot reduce the number of required steps against differential attacks, it reveals the fine-grained security of Pholkos against differential attacks. We believe that these results are helpful for estimating the security against known-key and hash function settings.

6.2 Related-Tweak Setting

In [13], the lower bounds for 4/7/5/6 or more steps for Pholkos-256-128/256-256/512/1024 was obtained indirectly by summing lower bounds of some previous steps, instead of using the exact MILP-aided approach, respectively. We reveal the lower bounds for the number of active S-boxes over all rounds for Pholkos-256-128/256-256 in the related-tweak setting. Our results updates the number of required steps for Pholkos-256-128 against differential attacks from 5 to 4. For Pholkos-1024, we update the number of lower bounds of active S-boxes in 5 steps.

6.3 Related-Tweakey Setting

In [13], the lower bounds for 4/5/4/4 or more steps for Pholkos-256-128/256-256/512/1024 was also derived by summing lower bounds of some previous steps. We show the lower bounds of Pholkos-256-128/256-256/1024 can be

Table 5 Lower bounds for the number of active S-boxes of Pholkos family over 1–10 steps. Bold is the minimum secure number of steps against differential attacks. Gray is the bounds derived for more steps from those results. Values in parentheses are the results of [13].

Setting	Variant	1S	2S	3S	4S	5S	6S	7S	8S	9S	10S
Single-key	Pholkos-256-128	5 (5)	25 (25)	35 (35)	**60** (**60**)	80 (80)	100 (100)	110 (110)	135 (135)	155 (140)	175 (160)
	Pholkos-256-256	5 (5)	25 (25)	35 (35)	**60** (**60**)	80 (80)	100 (100)	110 (110)	135 (135)	155 (140)	175 (160)
	Pholkos-512	5 (5)	25 (25)	**45** (**45**)	80 (80)	130 (130)	150 (150)	170 (170)	205 (205)	255 (210)	275 (230)
	Pholkos-1024	5 (5)	25 (25)	35 (35)	**70** (**70**)	105 (105)	165 (165)	240 (240)	285 (245)	290 (265)	320 (275)
Related-tweak	Pholkos-256-128	2 (2)	20 (20)	35 (35)	**49** (40)	63 (**55**)	77 (70)	91 (75)	105 (90)	- (105)	- (110)
	Pholkos-256-256	1 (1)	10 (10)	22 (22)	32 (32)	**45** (**45**)	54 (54)	64 (55)	75 (67)	86 (77)	- (90)
	Pholkos-512	4 (4)	25 (25)	**45** (**45**)	80 (80)	125 (84)	150 (104)	170 (125)	- (160)	- (164)	- (185)
	Pholkos-1024	5 (5)	25 (25)	35 (35)	**70** (**70**)	105 (87)	- (95)	- (112)	- (140)	- (157)	- (174)
Related-tweakey	Pholkos-256-128	0 (0)	8 (8)	22 (22)	32 (22)	**45** (30)	54 (**44**)	64 (44)	75 (52)	- (66)	- (66)
	Pholkos-256-256	0 (0)	4 (4)	14 (17)	26 (26)	38 (26)	**49** (34)	59 (**43**)	- (52)	- (52)	- (60)
	Pholkos-512	0 (0)	24 (24)	**45** (**45**)	- (48)	- (69)	- (90)	- (93)	- (114)	- (135)	- (138)
	Pholkos-1024	0 (0)	25 (10)	35 (35)	**70** (35)	105 (**45**)	- (70)	- (70)	- (80)	- (105)	- (105)

Table 6 Example of active input bytes of minimum #active S-boxes in 3 steps for Pholkos-256-256 in the related-tweakey setting.

Substate	Input bytes
P[0]	00001010 00000000
P[1]	01000001 01000000
K[0]	00001010 00000000
K[1]	01000001 01000000
T[0]	00001010 00000000
T[1]	01000001 01000000

found up to 8/7/5 steps, respectively. As a result, we successfully update the number of required steps for Pholkos-256-128/256-256/1024 against differential attacks from 6/7/5 steps to 5/6/4 steps, respectively. Besides, we reveal that the number of active S-boxes of 3 steps of Pholkos-256-256 is smaller than the results reported in [13]. Table 6 shows an example of the active input bytes in 3 steps for Pholkos-256-256, where 0 and 1 are defined as a non-active byte and an active byte, respectively.

7. Conclusion

In this paper, we developed efficient search methods based on MILP to obtain tighter lower bounds for the number of active S-boxes in a larger number of steps. For the first time, we derived the exact minimum number of active S-boxes of each variant up to the steps where the security against differential attacks can be ensure in related-tweak and related-tweakey settings. Our results indicate that Pholkos-256-128/256-256/512/1024 is secure after 4/5/3/4 steps in the related-tweak setting, and after 5/6/3/4 steps in the related-tweakey setting, respectively. Our results enable reducing the required number of steps to be secure against differential attacks of Pholkos-256-256 in related-tweak setting, and Pholkos-256-128/256 and Pholkos-1024 in the related tweakey setting by one step, respectively.

Acknowledgments

This research comprises a part of the results of "Research and Development of New Generation Cryptography for Secure Wireless Communication Services," which was commissioned by the Ministry of Internal Affairs and Communications of Japan under the "Research and Development for Expansion of Radio Resources" grant (grant no.JPJ000254)". This work was supported by JST PRESTO (grant no. JPMJPR2031).

References

[1] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," J. Cryptology, vol.4, no.1, pp.3–72, 1991.

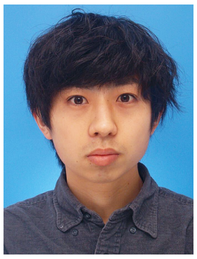
[2] E. Biham, "New types of cryptanalytic attacks using related keys," J. Cryptology, vol.7, no.4, pp.229–246, 1994.

[3] L.R. Knudsen, "Truncated and higher order differentials," Fast Software Encryption: Second International Workshop, Leuven, Belgium,

Proceedings, B. Preneel, ed., Lecture Notes in Computer Science, vol.1008, pp.196–211, Springer, 1994.
[4] E. Biham and A. Shamir, "Differential cryptanalysis of the full 16-round DES," Advances in Cryptology - CRYPTO'92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, Proceedings, E.F. Brickell, ed., Lecture Notes in Computer Science, vol.740, pp.487–496, Springer, 1992.
[5] N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, Revised Selected Papers, C. Wu, M. Yung, and D. Lin, eds., Lecture Notes in Computer Science, vol.7537, pp.57–76, Springer, 2011.
[6] Z. Xiang, W. Zhang, Z. Bao, and D. Lin, "Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers," Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, Proceedings, Part I, J.H. Cheon and T. Takagi, eds., Lecture Notes in Computer Science, vol.10031, pp.648–678, 2016.
[7] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, and L. Song, "Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers," Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., Proceedings, Part I, P. Sarkar and T. Iwata, eds., Lecture Notes in Computer Science, vol.8873, pp.158–178, Springer, 2014.
[8] Y. Sasaki and Y. Todo, "New impossible differential search tool from design and cryptanalysis aspects: Revealing structural properties of several ciphers," Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, Proceedings, Part III, J. Coron and J.B. Nielsen, eds., Lecture Notes in Computer Science, vol.10212, pp.185–215, 2017.
[9] L. Sun, W. Wang, and M. Wang, "More accurate differential properties of LED64 and Midori64," IACR Trans. Symmetric Cryptol., vol.2018, no.3, pp.93–123, 2018.
[10] 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.
[11] J. Guo, G. Liu, L. Song, and Y. Tu, "Exploring SAT for cryptanalysis: (Quantum) collision attacks against 6-round SHA-3," Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, Proceedings, Part III, S. Agrawal and D. Lin, eds., Lecture Notes in Computer Science, vol.13793, pp.645–674, Springer, 2022.
[12] 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.
[13] J. Bossert, E. List, S. Lucks, and S. Schmitz, "Pholkos – Efficient large-state tweakable block ciphers from the AES round function," Topics in Cryptology - CT-RSA 2022 - Cryptographers' Track at the RSA Conference 2022, Virtual Event, Proceedings, S.D. Galbraith, ed., Lecture Notes in Computer Science, vol.13161, pp.511–536, Springer, 2022.
[14] G.O. Inc., "Gurobi optimizer 6.5," Official webpage, http://www.gurobi.com/, 2015.

Nobuyuki Takeuchi received the B.E. and M.E. degree from University of Hyogo, Japan, in 2021 and 2023, respectively. His research interest had been cryptography. He has worked at SECOM CO., LTD. from 2023.

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.



Takuro Shiraya 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.

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. His current research interests include information security and cryptography.