Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 169638, 9 pages
http://dx.doi.org/10.1155/2013/169638

Research Article

Security Analysis of Scalable Block Cipher PP-1 Applicable to Distributed Sensor Networks

Yuseop Lee,[1] Kitae Jeong,[1] Jaechul Sung,[2] Changhoon Lee,[3] Seokhie Hong,[1] and Ku-Young Chang[4]

[1] *Center for Information Security Technologies (CIST), Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Republic of Korea*
[2] *Department of Mathematics, University of Seoul, Jeonnong-dong, Dongdaemun-gu, Seoul 130-743, Republic of Korea*
[3] *Department of Computer Science and Engineering, Seoul National University of Science and Technology, 232 Gongneung-ro, Nowon-gu, Seoul 139-743, Republic of Korea*
[4] *Electronics and Telecommunication Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 305-700, Republic of Korea*

Correspondence should be addressed to Changhoon Lee; chlee@seoultech.ac.kr

Received 12 August 2013; Accepted 22 August 2013

Academic Editor: Jongsung Kim

PP-1 is a scalable block cipher which can be implemented on a platform with limited resource. In this paper, we analyze the security of PP-1 by using truncated differential cryptanalysis. As concrete examples, we consider four versions of PP-1, PP-1/64, PP-1/128, PP-1/192, and PP-1/256. Our attack is applicable to full-round versions of them, respectively. The proposed attacks can recover a secret key of PP-1 with the computational complexity which is faster than the exhaustive search. These are the first known cryptanalytic results on PP-1.

1. Introduction

Recently, the research on lightweight block ciphers has received considerable attention. Since these can be efficiently implemented under restricted resources such as low-cost, low-power, and lightweight platforms, they are applicable to low-end devices such as RFID tags, sensor nodes, and smart devices [1–6]. So far, many lightweight block ciphers (e.g., HIGHT [7], CLEFIA [8], KATAN/KTANTAN [9], PRINTCIPHER [5], and PP-1 [10]) have been proposed.

PP-1 is an involutional SPN block cipher which can be implemented on a platform with limited resources. It supports the scalability, which allows using different data block sizes and secret key sizes. In detail, PP-1 is an n-bit scalable block cipher and supports $n/2n$-bit secret keys. ($n = 64, 128, 192, \ldots$). It uses an 8×8 S-box which is an involution and a bit-oriented permutation which is also an involution. As a result, it is a totally involutional cipher. To our knowledge, there is no cryptanalytic result on PP-1.

In this paper, we analyze the security of PP-1 on truncated differential cryptanalysis. As concrete examples, we consider four versions of PP-1, PP-1/64, PP-1/128, PP-1/192, and PP-1/256. Here, 64, 128, 192, and 256 indicate the length of data blocks. Our attack is applicable to full-round versions of them, respectively. Our attack results are summarized in Table 1. Here, PP-1/n_k means an n-bit PP-1 which supports a k-bit secret key. Note that since our attacks do not use the property of the key schedule of PP-1, the data complexity and the memory complexity of the attacks on PP-1/n_k and PP-1/n_2k have the same value. From this table, our attacks can recover a secret key of PP-1 with the computational complexity which is faster than the exhaustive search. These results are the first known cryptanalytic results on PP-1.

The rest of this paper is organized as follows. In Section 2, we briefly present PP-1. In Section 3, differentials on PP-1 are derived, and their probabilities are computed. Truncated differential cryptanalysis on each version of PP-1 is proposed in Sections 4, 5, and 6, respectively. Finally, we give our conclusion in Section 7.

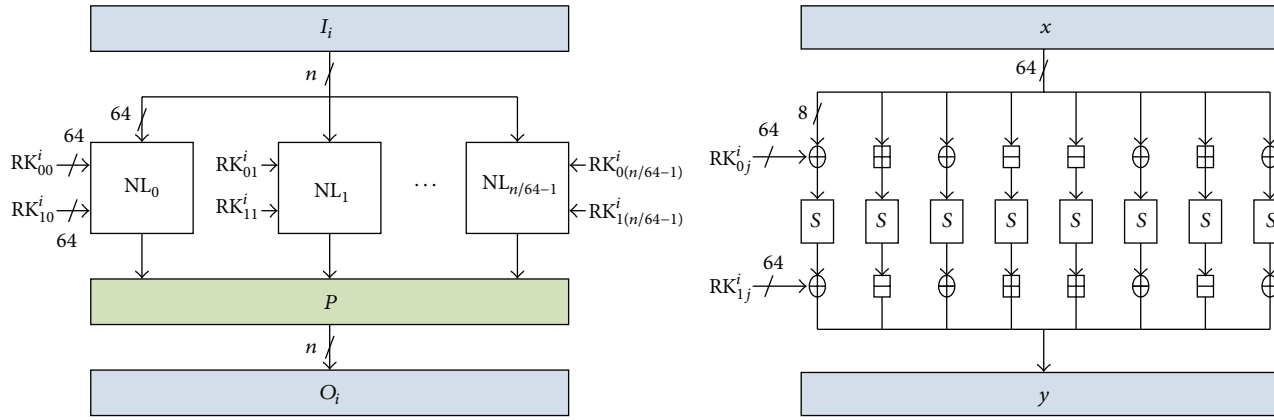

Figure 1: (a) The round function of PP-1 and (b) the nonlinear function NL_j.

Table 1: Our attack results on PP-1.

Target algorithm	Data complexity	Memory complexity	Computational complexity
PP-1/64_64	$2^{45.29}$ CP	$2^{41.29}$ bytes	$2^{45.29}$ encryptions
PP-1/64_128			$2^{48.21}$ encryptions
PP-1/128_128	$2^{103.45}$ CP	$2^{44.45}$ bytes	$2^{103.45}$ encryptions
PP-1/128_256			2^{168} encryptions
PP-1/192_192	$2^{157.85}$ CP	$2^{35.44}$ bytes	$2^{157.85}$ encryptions
PP-1/192_384			2^{296} encryptions
PP-1/256_256	$2^{210.84}$ CP	$2^{24.84}$ bytes	$2^{210.84}$ encryptions
PP-1/256_512			2^{432} encryptions

PP-1/n_k: an n-bit PP-1 with a k-bit secret key.
CP: chosen plaintexts.

2. Description of PP-1

PP-1 is an n-bit scalable block cipher and has r-round SPN structure ($n = 64, 128, 192, \ldots$). The length of a secret key is n or $2n$ bits. In [10], the designers of PP-1 proposed the following four versions of PP-1 as concrete examples.

(i) PP-1/64: $n = 64$, $r = 11$, that is, a 64-round block cipher with 64/128 bit secret keys and 11 rounds.

(ii) PP-1/128: $n = 128$, $r = 22$, PP-1/192: $n = 192$, $r = 32$, PP-1/256: $n = 256$, $r = 43$.

For the simplicity of notations, we denote an n-bit PP-1 with a k-bit secret key PP-1/n_k. And input/output values and a round key in round i are denoted by I^i, O^i, and RK^i, respectively ($i = 1, \ldots, r$).

We omit the key schedule of PP-1, as it is not effectively used in our attack.

2.1. The Round Function. As shown in Figure 1(a), the round function of PP-1 consists of (n/64) nonlinear NL functions ($NL_0, \ldots, NL_{(n/64-1)}$) and an n-bit involutional permutation P. For example, when $n = 64$, the round function uses only NL_0. Note that P is not conducted in the last round. ($NL_0, \ldots, NL_{(n/64-1)}$) have the same structure but different round keys are used.

In round i, two n-bit round keys $RK^i = (RK^i_0, RK^i_1)$ are used as follows:

$$RK^i_0 = RK^i_{00} \| \cdots \| RK^i_{0(n/64-1)},$$
$$RK^i_1 = RK^i_{10} \| \cdots \| RK^i_{1(n/64-1)}. \quad (1)$$

Here, NL_j takes 128 bit sub-round key (RK^i_{0j}, RK^i_{1j}) ($j = 0, \ldots, (n/64 - 1)$).

2.2. The Nonlinear Function NL. In round i, NL_j outputs a 64 bit value from a 64 bit input value and a 128 bit subround key (RK^i_{0j}, RK^i_{1j})($j = 0, \ldots, (n/64-1)$). It consists of one 8×8 S-box S, XOR($\oplus$), addition($\boxplus$), and substraction($\boxminus$) modulo 2^8 (see Figure 1(b)).

A 128 bit sub-round key (RK^i_{0j}, RK^i_{1j}) is divided into eight 8 bit elementary keys as follows, respectively:

$$RK^i_{0j} = RK^i_{0j0} \| \cdots \| RK^i_{0j7},$$
$$RK^i_{1j} = RK^i_{1j0} \| \cdots \| RK^i_{1j7}. \quad (2)$$

Thus, each elementary key is XORed or added or subtracted with an 8 bit intermediate value. For example, RK^i_{1j0} is XORed with the 8 bit output value of the first S-box.

2.3. The Permutation Function P. P is an n-bit involutional bit-oriented permutation. It is constructed by using two algorithms, the auxiliary algorithm (Algorithm 1) to compute auxiliary permutation Prm and the main algorithm (Algorithm 2) to compute permutation P.

For example, the 128 bit permutation P is obtained as a result of 64 calls of Algorithm 2 for a pair numbered as pno from 1 to 64, the number of block bits $nBb = 128$ and the number of S-box bits $nSb = 8$. When $pno = 2$, the value y of Prm is equal to 9 and the resultant pair $(px, py) = (3, 18)$. It means that the third bit of the input value is mapping to the eighteenth bit of the output value.

(1) $nS \leftarrow nBb \text{ div } nSb$.
(2) $Sno \leftarrow (x \bmod nS) + 1$.
(3) $Sb \leftarrow ((x-1) \text{ div } nS) + 1$.
(4) $y \leftarrow (Sno - 1) \cdot nSb + Sb$.
(5) Return y.

Algorithm 1: Prm(x, nBb, nSb).

(1) $y \leftarrow \text{Prm}(pno, nBb \text{ div } 2, nSb \text{ div } 2)$.
(2) $px \leftarrow 2 \cdot pno - 1$.
(3) $py \leftarrow 2 \cdot y$.
(4) Return (px, py).

Algorithm 2: $P(pno, nBb, nSb)$.

3. Construction of Differentials on PP-1

In this section, we introduce the methodology of constructing differentials on PP-1 used in our attacks. For the simplicity of notations, we define the following notations.

(i) $(\alpha \rightarrow \beta)_t$: a t-round differential characteristic where input/output differences are α and β, respectively.

(ii) $[\alpha \rightarrow \beta]_t$: a t-round differential where input/output differences are α and β, respectively.

(iii) (a, x): a byte string where the ath byte value is x and the other bytes are zero (the index of the left most byte is 0).

3.1. Differential Characteristic on PP-1. In general, a differential characteristic with the higher probability passes less nonlinear operations, such as S-box, than it with the lower probability. Recall that a *NF*-function consists of S-box, addition, substraction, and XOR. Among these operations, nonlinear operations are S-box, addition, and subtraction. Thus, in order to construct a differential characteristic with a high probability, we should avoid them.

We examined such differential characteristics on PP-1. As a result, we found several t-round differential characteristics with a probability of $2^{-7 \cdot t}$. For example, in the case of PP-1/64, we can construct $((7, \texttt{0x01}) \rightarrow (7, \texttt{0x01}))_t$ with a probability of $2^{-7 \cdot t}$. This characteristic passes only one S-box in each round and the probability that S-box outputs an output difference 0x01 from an input difference 0x01 is 2^{-7}.

We expect that this type of difference characteristics have the highest probability. That is, they pass least S-boxes, addition operations, and subtraction operations. We extend it to differentials in the next subsection.

3.2. Finding of Differentials with a High Probability. The probability of a differential is computed by adding the probabilities of all differential characteristics which are included in it. The more differential characteristics with a high probability a differential includes, the higher its probability is. Thus, in order to find a differential on PP-1 with a high probability, we consider the following criteria.

(i) In each round, a differential characteristic has only one active S-box.

(ii) In each round, a differential characteristic does not pass addition/subtraction operations.

The probabilities that all t-round differential characteristics satisfying the above criteria are at least $2^{-7 \cdot t}$, since the minimum probability from the difference distribution table on S-box is 2^{-7}. Thus, we measure the probability of a differential by counting only the number of differential characteristics which satisfy the above criteria and are included in it. That is, if there are w such differential characteristics, the probability of a differential including them is at least $w \cdot 2^{-7 \cdot t}$. On the other hand, in the case of differential characteristics which do not satisfy the above criteria, they pass the additional nonlinear operations in each round. In this case, the probabilities of them are much smaller than $2^{-7 \cdot t}$. Thus, we expect that differential characteristics which do not satisfy the above criteria depend on the probability of a differential less.

In order to count efficiently differential characteristics satisfying the above criteria, we consider differences δ's satisfying the following conditions.

(i) $\delta = (a, x)$ where $(a \bmod 8) \in \{0, 2, 5, 7\}$ and x is a nonzero byte value.

(ii) $P(\delta) = (b, y)$ where $(b \bmod 8) \in \{0, 2, 5, 7\}$ and y is a nonzero byte value.

Let $\mathcal{D}$ be a set containing such δ's. We can easily prove that all t-round differential characteristics satisfying the above criteria include $(\delta_i \rightarrow \delta_j)_1$ in each round $(\delta_i, \delta_j \in \mathcal{D})$. It means that we only need to consider $\mathcal{D}$ in order to find all differential characteristics holding the above criteria.

Let $N(\delta_i, \delta_j, t)$ be the number of $(\delta_i \rightarrow \delta_j)_t$. For each δ_i and δ_j included in $\mathcal{D}$, we compute w using the following recurrence relation:

$$N\left(\delta_i, \delta_j, t+1\right) = \sum_{\delta_k \in \mathcal{D}} N\left(\delta_i, \delta_k, 1\right) \cdot N\left(\delta_k, \delta_j, t\right). \quad (3)$$

Since $N(\delta_i, \delta_k, 1)$ is compute by using the difference distribution table for an S-box of PP-1, we can easily compute $N(\delta_i, \delta_j, t)$ for given t.

For example, we found twenty one δ's for PP-1/64 (see Table 2). As a simulation result, $N((7, \texttt{0x01}), (7, \texttt{0x09}), 8)$ is 8429. It means that the probability of $[(7, \texttt{0x01}) \rightarrow (7, \texttt{0x09})]_8$ is $2^{-42.96} (= 8429 \cdot 2^{-7 \cdot 8})$.

4. Truncated Differential Analysis on PP-1/64

In this section, we propose truncated differential analysis on full-round PP-1/64_64 and full-round PP-1/64_128. Since PP-1/64_64 and PP-1/64_128 have the same structure except the key schedule, the attack procedures on them are similar. Thus, we mainly introduce the attack procedure on PP-1/64_64.

Table 2: A difference set $\mathscr{D}$ for PP-1/64.

(0, 0x20)	(0, 0x04)	(0, 0x01)	(2, 0x40)	(2, 0x10)	(2, 0x20)	(2, 0x30)
(5, 0x02)	(5, 0x04)	(5, 0x80)	(5, 0x84)	(5, 0x01)	(5, 0x08)	(5, 0x09)
(7, 0x02)	(7, 0x04)	(7, 0x80)	(7, 0x84)	(7, 0x01)	(7, 0x08)	(7, 0x09)

By using the method in the previous section, we construct forty nine 8-round differentials $[P((7,\alpha)) \rightarrow (7,\beta)]_8$ ($\alpha, \beta \in$ {0x01, 0x02, 0x04, 0x08, 0x09, 0x80, 0x84}). And we extend these 8-round differentials to total 1785(= $7 \cdot 255$) 9-round differentials $[P((7,\alpha)) \rightarrow P((7, x_j^i))]_9$ ($x_j^i \in X^i$). Here, X^i's are defined as follows ($i = 1, \ldots, 6$):

$$
\begin{aligned}
X^0 &= \{00000000_2\},\\
X^1 &= \{0000?00?_2 \mid ? \in \{0,1\}\} - X^0,\\
X^2 &= \{0?00?00?_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{1} X^i,\\
X^3 &= \{??00??0?_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{2} X^i,\\
X^4 &= \{??00????_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{3} X^i,\\
X^5 &= \{???0????_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{4} X^i,\\
X^6 &= \{????????_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{5} X^i.
\end{aligned}
\tag{4}
$$

4.1. Construction of Structures. We consider a structure S_i consisting of 256 plaintext; that is, $S_i = \{(i \parallel j) \mid j = 0, 1, 2, \ldots, 255\}$ where i is a 56 bit fixed value. Then we can compose 2^{15} plaintext pairs for each structure. Among these plaintext pairs, there are 2^7 plaintext pairs where an input difference of round 2 is one of $P((7,\alpha))$ for each α. Table 3 presents the expected number of right plaintext pairs where an output difference of round 10 is included in $P((7, X^i))$. These values are computed as follows:

$$
\sum_{\alpha} \sum_{x_j^i \in X^i} 2^7 \cdot \Pr\left[\left[\alpha \longrightarrow x_j^i\right]_9\right]. \tag{5}
$$

4.2. Truncated Differential Analysis on PP-1/64. The main idea of our attack is to exploit the fact that the expected number of plaintext pairs where an output difference of round 10 is included in $P(X^i)$ is $2^{-30.66} \sim 2^{-35.29}$ for each structure (see Table 3).

In our attack on PP-1/64_64, we first obtain a 72 bit partial information on $\mathrm{RK}^{11} = (\mathrm{RK}_0^{11}, \mathrm{RK}_1^{11})$ and an 8 bit RK_{007}^1. The attack procedure is as follows (see Figure 2).

(1) Choose $2^{37.29}$ structures which are composed of 256 plaintexts and obtain the corresponding ciphertexts. From these ciphertexts, compute $2^{52.29}(= 2^{15} \cdot 2^{37.29})$ ciphertext pairs (C^i, C^{i*}) (note that we can compute total 2^{15} ciphertext pairs for each structure).

(2) Check that the difference ΔC^i between ciphertext pair (C^i, C^{i*}) is 0x????00??00??????; that is, $\Delta C_{02}^i = \Delta C_{04}^i =$ 0x00 for each $i(? \in \{0,1\}^4)$. We keep all ciphertext pairs passing Step (2) and the corresponding plaintext pairs in a table and call a set containing them $\mathscr{A}$.

(3) Filter out the ciphertext pairs where ΔC_{00}^i, ΔC_{01}^i, ΔC_{03}^i, ΔC_{05}^i and ΔC_{06}^i are not zero in $\mathscr{A}$. Do the following for the remaining ciphertext pairs:

  (a) Guess an 8 bit RK_{107}^{11} (note that this substep indicates "Guess 1" in Figure 2).

  (b) Partially encrypt all remaining ciphertext pairs with the guessed round key RK_{107}^{11} to get ΔI_{07}^{11}. Check that ΔI_{07}^{11} is included in $P(X^1)$ from (4). If it is included in $P(X^1)$, add the counter, corresponding to the guessed key, to one.

  (c) Output a guessed key which has the maximal counter as a right RK_{107}^{11}.

(4) From $\mathscr{A}$, filter out the ciphertext pairs where $\Delta C_{00}^i, \Delta C_{03}^i$, ΔC_{05}^i, and ΔC_{06}^i are not zero and the ciphertext pairs considered in Step (3). Do the following for the remaining ciphertext pairs:

  (a) Check that ΔC_{07} is a nonzero value for each remaining ciphertext pair. Partially encrypt the ciphertext pairs passing this test with the recovered round key RK_{107}^{11} to obtain ΔI_{07}^{11}. If this value is not included in $P(X^1)$, filter out the corresponding ciphertext pairs.

  (b) Guess 16 bit round keys $(\mathrm{RK}_{001}^{11}, \mathrm{RK}_{101}^{11})$. ("Guess 2" in Figure 2).

  (c) Similarly to Step (3)(b), partially encrypt the remaining ciphertext pairs with the guessed round keys to get ΔI_{01}^{11}. Check that ΔI_{01}^{11} is

Table 3: The expected number of right pairs for PP-1/64.

Set of output differences of round 10	The expected number of right pairs
$P((7, X^1))$	$2^{-35.29}$
$P((7, X^2))$	$2^{-35.27}$
$P((7, X^3))$	$2^{-33.19}$
$P((7, X^4))$	$2^{-32.65}$
$P((7, X^5))$	$2^{-31.53}$
$P((7, X^6))$	$2^{-30.66}$

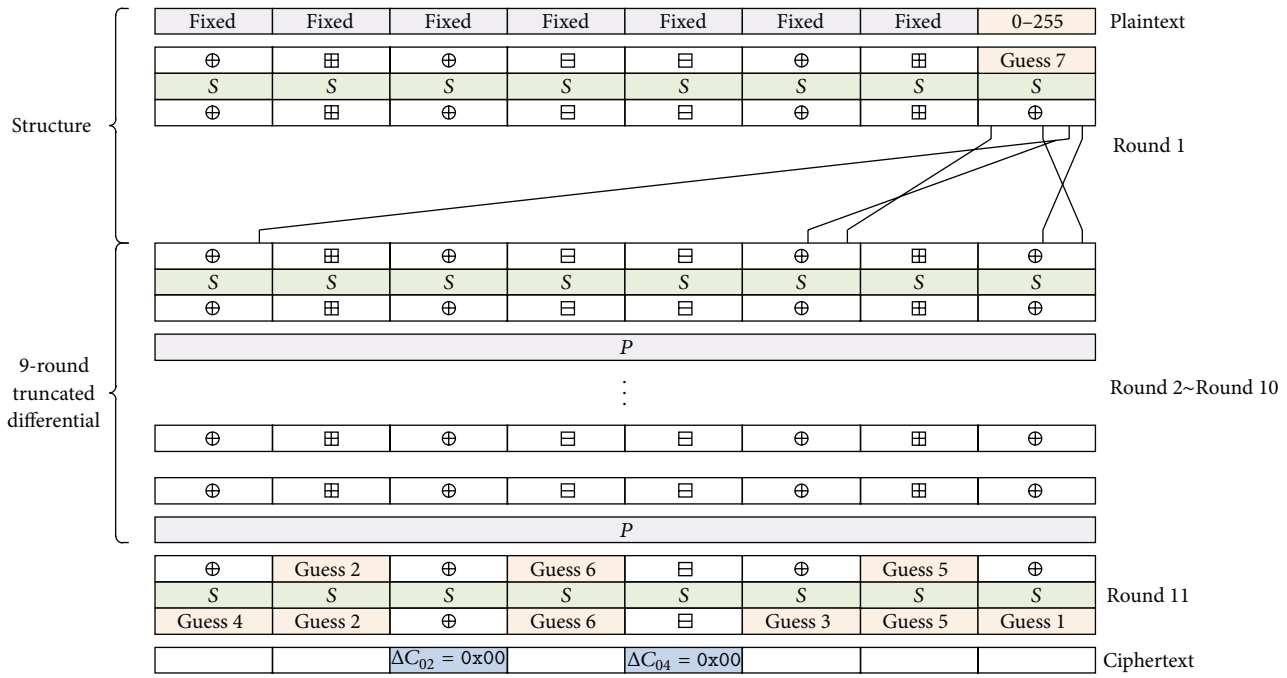

Figure 2: The attack procedure on full-round PP-1/64_64.

included in $P(X^2)$ from (4). If it is included in $P(X^2)$, add the counter, corresponding to the guessed key, to one.

(d) Output a guessed key which has the maximal counter as right RK^{11}_{001} and RK^{11}_{101}.

(5) From $\mathscr{A}$, discard the ciphertext pairs where ΔC^i_{00}, ΔC^i_{03}, and ΔC^i_{06} are not zero and the ciphertext pairs considered in Step (3) and (4). Do the following for the remaining ciphertext pairs:

(a) Similarly to Step (4)(a), filter out the ciphertext pairs where ΔI^{11}_{07} and ΔI^{11}_{01} are not included in $P(X^1)$ and $P(X^2)$, respectively.

(b) Guess an 8 bit round keys RK^{11}_{105}. ("Guess 3" in Figure 2).

(c) Similarly to Step (4)(c), check that ΔI^{11}_{05} is included in $P(X^3)$. Output a guessed key which has the maximal counter as a right RK^{11}_{105}.

(6) From $\mathscr{A}$, discard the ciphertext pairs where ΔC^i_{03} and ΔC^i_{06} are not zero and the ciphertext pairs considered in Step (3), (4) and (5). Do the following for the remaining ciphertext pairs:

(a) Similarly to Step (5)(a), filter out the ciphertext pairs where ΔI^{11}_{07}, ΔI^{11}_{01} and ΔI^{11}_{05} are not included in $P(X^1)$, $P(X^2)$ and $P(X^3)$, respectively.

(b) Guess an 8 bit round keys RK^{11}_{100}. ("Guess 4" in Figure 2).

(c) Similarly to Step (5)(c), check that ΔI^{11}_{00} is included in $P(X^4)$. Output a guessed key which has the maximal counter as a right RK^{11}_{100}.

(7) From $\mathscr{A}$, filter out the ciphertext pairs where ΔC^i_{03} is not zero and the ciphertext pairs considered in Step (3), (4), (5) and (6). Do the following for the remaining ciphertext pairs:

(a) Similarly to Step (6)(a), filter out the ciphertext pairs where ΔI^{11}_{07}, ΔI^{11}_{01}, ΔI^{11}_{05} and ΔI^{11}_{00} are not included in $P(X^1)$, $P(X^2)$, $P(X^3)$ and $P(X^4)$, respectively.

(b) Guess 16 bit round keys (RK^{11}_{006}, RK^{11}_{106}). ("Guess 5" in Figure 2).

(c) Similarly to Step (6)(c), check that ΔI^{11}_{06} is included in $P(X^5)$. Output a guessed key which has the maximal counter as right RK^{11}_{006} and RK^{11}_{106}.

(8) From $\mathscr{A}$, filter out the ciphertext pairs considered in Step (3), (4), (5), (6) and (7). Do the following for the remaining ciphertext pairs:

(a) Similarly to Step (7)(a), filter out the ciphertext pairs where ΔI^{11}_{07}, ΔI^{11}_{01}, ΔI^{11}_{05}, ΔI^{11}_{00} and ΔI^{11}_{06} are not included in $P(X^1)$, $P(X^2)$, $P(X^3)$, $P(X^4)$ and $P(X^5)$, respectively.

(b) Guess 16 bit round keys (RK^{11}_{003}, RK^{11}_{103}). ("Guess 6" in Figure 2).

(c) Similarly to Step (7)(c), check that ΔI_{03}^{11} is included in $P(X^6)$. Output a guessed key which has the maximal counter as right RK_{003}^{11} and RK_{103}^{11}.

(9) Get the the corresponding plaintext pairs to all ciphertext pairs considered in Step (3)(b), (4)(c), (5)(c), (6)(c), (7)(c) and (8)(c), respectively. With them, do the following:

(a) Guess an 8 bit RK_{007}^{1} ("Guess 7" in Figure 2).
(b) Partially encrypt the plaintext pairs in Step (9) with the guessed round key to obtain the output difference of the 8th S-box in round 1, that is, $(P^{-1}(I^2))_{07}$.
(c) If the computed difference is included in $\{\mathtt{0x01}, \mathtt{0x02}, \mathtt{0x04}, \mathtt{0x08}, \mathtt{0x09}, \mathtt{0x80}, \mathtt{0x84}\}$, add the counter, corresponding to the guessed key, to one.
(d) Output a guessed key which has the maximal counter as a right RK_{007}^{1}.

(10) With an 80 bit suggested round key, compute a secret key by operating the key schedule of PP-1_64. Output the computed secret key as a right secret key of PP-1_64.

In our attack on PP-1/64_64, we construct $2^{37.29}$ structures which are composed of 256 plaintexts. Thus, the data complexity of our attack is about $2^{45.29}(\approx 2^{37.29} \cdot 2^8)$ chosen plaintexts. We store all ciphertext pairs passing Step (2) and the corresponding plaintext pairs in a table. The probability that a ciphertext pair passes Step (2) is 2^{-16}. Thus, $2^{36.29}(\approx 2^{52.29} \cdot 2^{-16})$ ciphertext pairs pass this step. Hence, the memory complexity of this attack is about $2^{41.29}(\approx 2^{52.29} \cdot 2^{-16} \cdot 4 \cdot 8)$ memory bytes.

The computational complexity of our attack is dominated by Step (1). The computational complexity of Step (1) is about $2^{45.29}(\approx 2^{37.29} \cdot 2^8)$ encryptions. The probability that a wrong ciphertext pair passes Step (3) is 2^{-40}. Since the expected number of the remaining wrong ciphertext pairs is $2^{-3.71}(\approx 2^{36.29} \cdot 2^{-40})$, we expect that only right ciphertext pairs are survived. From (4), we can check easily that all ciphertext pairs where the corresponding ΔO^{10}'s are included in $P(X^1)$ pass Step (3). Thus, the expected number of right ciphertext pairs is $4(\approx 2^{37.29} \cdot 2^{-35.29})$ from Table 3. The computational complexity of Step (3)(b) is about $2^{3.54}(\approx 2^8 \cdot 4 \cdot 1/11 \cdot 1/8)$ encryptions. Similarly to Step (3)(b), the computational complexities of other steps are also small. Hence, the computational complexity of our attack on PP-1/64_64 is about $2^{45.29}$ encryptions.

In the case of the attack on PP-1/64_128, the data and memory complexities are the same as them of the attack on PP-1/64_64. However, the computational complexity of this attack is dominated by Step (1) and (10), since we should do an exhaustive search for the remaining 48 bit key information. The computational complexity of Step (1) is about $2^{45.29}(\approx 2^{37.29} \cdot 2^8)$ encryptions. In Step (10), the probability that a wrong key passes this step is 2^{-64}. Thus, it is sufficient to use just one plaintext/ciphertext pair. The computational complexity of Step (10) is about 2^{48}. Hence, the computational complexity of PP-1/64_128 is about $2^{48.21}(\approx 2^{45.29} + 2^{48})$ encryptions.

5. Truncated Differential Analysis on Full-Round PP-1/128

Our attacks on full-round PP-1/128_128 and full-round PP-1/128_256 use 20-round differentials which are constructed by using the method introduced in Section 3. In detail, in order to construct them, we consider twenty five 19-round differentials $[P((15, \alpha)) \rightarrow (15, \beta)]_{19}$ $(\alpha, \beta \in \{\mathtt{0x01}, \mathtt{0x08}, \mathtt{0x09}, \mathtt{0x10}, \mathtt{0x80}\})$. Then we extend these 19-round differentials to total $1275(= 5 \cdot 255)$ 20-round differentials $[P((15, \alpha)) \rightarrow P((15, y_j^i))]_{20}$ $(y_j^i \in Y^i)$. Here, Y^i's are defined as follows $(i = 1, \ldots, 7)$:

$$
\begin{aligned}
Y^0 &= \{00000000_2\},\\
Y^1 &= \{0000?00?_2 \mid ? \in \{0,1\}\} - Y^0,\\
Y^2 &= \{0?00?00?_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{1} Y^i,\\
Y^3 &= \{??00?00?_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{2} Y^i,\\
Y^4 &= \{??00??0?_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{3} Y^i,\\
Y^5 &= \{??00????_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{4} Y^i,\\
Y^6 &= \{???0????_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{5} Y^i,\\
Y^7 &= \{????????_2 \mid ? \in \{0,1\}\} - \bigcup_{i=0}^{6} Y^i.
\end{aligned}
\tag{6}
$$

In the similar manner to the previous section, we choose a structure S_i which consist of 256 plaintext; that is, $S_i = \{(i \parallel j) \mid j = 0, 1, 2, \ldots, 255\}$, where i is a 120 bit fixed value. Then, as shown in Table 4, we can calculate the expected number of right plaintext pairs where ΔO^{21} is included in $P(15, (Y^i))$ for each structure.

5.1. Truncated Differential Analysis on PP-1/128. Our attack on full-round PP-1/128_128 is similar to that on full-round PP-1/128_256. Thus, we mainly present the attack procedure on PP-1/128_128. Since it is similar to the attack procedure on full-round PP-1/64_64, we briefly introduce it. The attack procedure on full-round PP-1/128_128 is as follows (see Figure 3).

(1) Select $2^{95.45}$ structures which are composed of 256 plaintexts and get the corresponding ciphertexts.

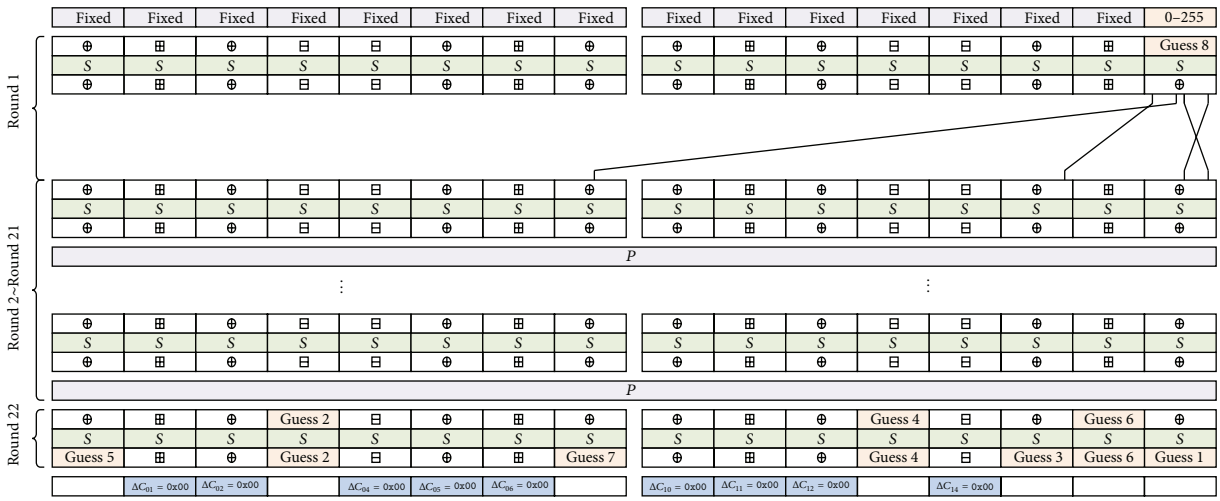

Figure 3: The attack procedure on full-round PP-1/128_128.

Table 4: The expected number of right pairs for PP-1/128.

Set of output differences of round 21	The expected number of right pairs
$P((15, Y^1))$	$2^{-93.33}$
$P((15, Y^2))$	$2^{-93.45}$
$P((15, Y^3))$	$2^{-92.64}$
$P((15, Y^4))$	$2^{-92.01}$
$P((15, Y^5))$	$2^{-96.86}$
$P((15, Y^6))$	$2^{-89.78}$
$P((15, Y^7))$	$2^{-88.92}$

From these ciphertexts, compute $2^{110.45}$ ciphertext pairs (C^i, C^{i*}).

(2) Check that $\Delta C^i_{01} = \Delta C^i_{02} = \Delta C^i_{04} = \Delta C^i_{05} = \Delta C^i_{06} = \Delta C^i_{10} = \Delta C^i_{11} = \Delta C^i_{12} = \Delta C^i_{14} = 0$ for each ciphertext pair. We keep all ciphertext pairs passing Step (2) and the corresponding plaintext pairs in a table and call a set containing them $\mathscr{A}$.

(3) From $\mathscr{A}$, filter out the ciphertext pairs where ΔC^i_{00}, ΔC^i_{03}, ΔC^i_{07}, ΔC^i_{13}, ΔC^i_{15}, and ΔC^i_{16} are not zero. Do the following for the remaining ciphertext pairs:

(a) Guess an 8 bit round key RK^{22}_{117} ("Guess 1" in Figure 3).

(b) Check that ΔI^{22}_{17} is included in $P(Y^1)$. Output a guessed key which has the maximal counter as right RK^{22}_{117}.

(4) Similarly to Step (3), determine sequentially the following right round keys.

(a) $(\mathrm{RK}^{22}_{003}, \mathrm{RK}^{22}_{103})$ ("Guess 2"),

(b) RK^{22}_{115} ("Guess 3").

(c) $(\mathrm{RK}^{22}_{013}, \mathrm{RK}^{22}_{115})$ ("Guess 4").

(d) RK^{22}_{101} ("Guess 5").

(e) $(\mathrm{RK}^{22}_{016}, \mathrm{RK}^{22}_{116})$ ("Guess 6").

(f) RK^{22}_{117} ("Guess 7").

(5) Get the the corresponding plaintext pairs to all ciphertext pairs considered in Steps (3) and (4). With them, do the following:

(a) Guess an 8 bit RK^1_{017} ("Guess 8").

(b) Partially encrypt the plaintext pairs in Step (5) with the guessed round key to obtain the output difference of the 8th S-box in second NF-function of round 1, that is, $(P^{-1}(I^2))_{17}$.

(c) If the computed difference is included in {0x01, 0x08, 0x09, 0x10, 0x80}, add the counter, corresponding to the guessed key, to one.

(d) Output a guessed key which has the maximal counter as a right RK^1_{017}.

(6) With an 88 bit suggested round key, do an exhaustive search for the remaining 40 bit key information by using one trial encryption. During this procedure, if a 128 bit secret key satisfies one known plaintext/ciphertext pair, output this 128 bit secret key as a right 128 bit secret key of full-round PP-1/128_128.

In this attack, we construct $2^{95.45}$ structures. Thus, the data complexity of our attack on full-round PP-1/128_128 is about $2^{103.45}(\approx 2^{95.45} \cdot 2^8)$ chosen plaintexts. In Step (2), since the probability that a ciphertext pair passes Step (2) is 2^{-72}, we store $2^{38.45}(\approx 2^{110.45} \cdot 2^{-72})$ ciphertext pairs pass this step and the corresponding plaintext pairs in a table. Thus, the memory complexity of this attack is about $2^{44.45}(\approx 2^{38.45} \cdot 4 \cdot 16)$ memory bytes. The computational complexity of this attack is dominated by Step (1), that is, about $2^{103.45}(\approx 2^{95.45} \cdot 2^8)$ encryptions.

In the case of the attack on full-round PP-1/128_256, the data and memory complexities are the same as them of the

attack on full-round PP-1/128_128. However, the computational complexity of this attack is dominated by Step (6), since we should do an exhaustive search for the remaining 168 bit key information. In Step (6), the probability that a wrong key that passes this step is 2^{-128}. Thus, this step needs two plaintext/ciphertext pairs. The computational complexity of Step (6) is about $2^{168}(\approx 2^{168} + 2^{168} \cdot 2^{-128})$ encryptions. Hence, the computational complexity of our attack on full-round PP-1/128_256 is about 2^{168} encryptions.

6. Truncated Differential Analysis on Full-Round PP-1/192 and PP-1/256

This section introduces our attack results on full-round PP-1/192 and full-round PP-1/256. Overall, the attack procedures on them are similar to the attack procedures on PP-1/64.

Our attacks on full-round PP-1/192 uses $2^{149.85}$ structures $S_i = \{(i \parallel j) \mid j = 0, 1, 2, \ldots, 255\}$, where i is a 184 bit fixed value. First, we construct thirty six 29-round differentials $[P((23, \alpha)) \rightarrow (23, \beta)]_{29}$ $(\alpha, \beta \in \{\texttt{0x01}, \texttt{0x02}, \texttt{0x08}, \texttt{0x09}, \texttt{0x40}, \texttt{0x80}\})$. Then we extend these 29-round differentials to total $1530(= 6 \cdot 255)$ 30-round truncated differentials $[P((23, \alpha)) \rightarrow P((23, y_j^i))]_{30}$ $(y_j^i \in Y^i)$. Note that Y^i's used in this attack are the same as them in the attack on full-round PP-1/128. Table 5 presents the expected number of right plaintext pairs where ΔO^{31} is included in $P(23, (Y^i))$ in each structure. The complexities of our attacks are as follows.

PP-1/192_192(PP-1/192_384)

(a) The data complexity: about $2^{157.85}$ chosen plaintexts.

(b) The memory complexity: about $2^{35.44}$ memory bytes.

(c) The computational complexity: about $2^{157.85}(2^{296})$ encryptions.

In the case of PP-1/256, we consider $2^{202.84}$ structure $S_i = \{(i \parallel j) \mid j = 0, 1, 2, \ldots, 255\}$, where i is a 248 bit fixed value. And we construct 204041-round differentials $[P((31, \alpha)) \rightarrow P((31, y_j^i))]_{41}$ $(\alpha \in \{\texttt{0x01}, \texttt{0x02}, \texttt{0x04}, \texttt{0x08}, \texttt{0x09}, \texttt{0x10}, \texttt{0x40}, \texttt{0x80}\})$ $(y_j^i \in Y^i)$. Note that Y^i's used in this attack are also the same as them in the attack on full-round PP-1/128. Table 6 presents the expected number of right plaintext pairs where ΔO^{42} is included in $P(31, (Y^i))$ in each structure. The complexities of our attacks are as follows.

PP-1/256_256(PP-1/256_512)

(a) The data complexity: about $2^{210.84}$ chosen plaintexts.

(b) The memory complexity: about $2^{24.84}$ memory bytes.

(c) The computational complexity: about $2^{210.84}(2^{432})$ encryptions.

Table 5: The expected number of right pairs for PP-1/192.

Set of output differences of round 31	The expected number of right pairs
$P((23, Y^1))$	$2^{-147.85}$
$P((23, Y^2))$	$2^{-147.01}$
$P((23, Y^3))$	$2^{-145.93}$
$P((23, Y^4))$	$2^{-145.48}$
$P((23, Y^5))$	$2^{-144.38}$
$P((23, Y^6))$	$2^{-143.26}$
$P((23, Y^7))$	$2^{-142.38}$

Table 6: The expected number of right pairs for PP-1/256.

Set of output differences of round 42	The expected number of right pairs
$P((31, Y^1))$	$2^{-200.73}$
$P((31, Y^2))$	$2^{-200.84}$
$P((31, Y^3))$	$2^{-199.89}$
$P((31, Y^4))$	$2^{-199.23}$
$P((31, Y^5))$	$2^{-198.10}$
$P((31, Y^6))$	$2^{-197.09}$
$P((31, Y^7))$	$2^{-196.18}$

7. Conclusion

In this paper, we have presented the first known cryptanalytic results of four concrete versions of a scalable block cipher PP-1, full-round PP-1/64, full-round PP-1/128, full-round PP-1/192, and full-round PP-1/256, by using truncated differential cryptanalysis. As summarized in Table 1, our attacks on these algorithms require computational complexities smaller than the exhaustive search. These results indicate that PP-1 is vulnerable to truncated differential cryptanalysis and that it is insecure.

Acknowledgments

This research was supported by the Ministry of Science, ICT and Future Planning (MSIP), Korea, under the Convergence Information Technology Research Center (C-ITRC) support Program (NIPA-2013-H0301-13-3007) supervised by the National IT Industry Promotion Agency (NIPA).

References

[1] J. Chen, B. Mariam, and M. Matsumoto, "A single mobile target tracking in voronoi-based clustered wireless sensor network," *Journal of Information Processing Systems*, vol. 1, pp. 17–28, 2011.

[2] C. Huang, R. H. Cheng, S. R. Chen, and C. Li, "Enhancing network availability by tolerance control in multi-sink wireless sensor network," *Journal of Convergence*, vol. 1, no. 1, pp. 15–22, 2010.

[3] P. Sarkar and A. Saha, "Security enhanced communication in wireless sensor networks using reed-muller codes and partially balanced incomplete block designs," vol. 2, pp. 23–30.

[4] D. Kumar, T. Aseri, and R. Patel, "Multi-hop communication routing (MCR) protocol for heterogeneous wireless sensor networks," *International Journal of Information Technology, Communications and Convergence*, vol. 1, pp. 130–145, 2010.

[5] H. Lim, K. Jang, and B. Kim, "A study on design and implementation of the ubiquitous computing environment-based dynamic smart on/off-line learner tracking system," *Journal of Information Processing Systems*, vol. 6, no. 4, pp. 609–620, 2010.

[6] B. Xie, A. Kumar, D. Zhao, R. Reddy, and B. He, "On secure communication in integrated heterogeneous wireless networks," *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 1, pp. 4–23, 2010.

[7] D. Hong, J. Sung, S. Hong et al., "HIGHT: a new block cipher suitable for low-resource device," *Cryptographic Hardware and Embedded Systems*, vol. 4249, pp. 46–59, 2006.

[8] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA (extended abstract)," *Fast Software Encryption*, vol. 4593, pp. 181–195, 2007.

[9] C. de Cannière, O. Dunkelman, and M. Knežević, "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers," *Cryptographic Hardware and Embedded Systems*, vol. 5747, pp. 272–288, 2009.

[10] K. Bucholc, K. Chmiel, A. Grocholewska-Czuryło, E. Idzikowska, I. Janicka-Lipska, and J. Stokłosa, "Scalable PP-1 block cipher," *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 2, pp. 401–411, 2010.