On the Pseudorandomness of the AES Finalists – RC6 and Serpent

Tetsu Iwata and Kaoru Kurosawa

Department of Electrical and Electronic Engineering,
Faculty of Engineering,
Tokyo Institute of Technology
2–12–1 O-okayama, Meguro-ku, Tokyo 152–8552, Japan
tez@ss.titech.ac.jp, kurosawa@ss.titech.ac.jp

Abstract. Luby and Rackoff idealized DES by replacing each *round function* with *one large* random function. In this paper, we introduce a primitive-wise idealization in which some of the *primitive operations* of the round function are left untouched and some of them are replaced with *small* random functions or permutations. We then prove that a four round primitive-wise idealized RC6 is not a pseudorandom permutation and a three round primitive-wise idealized Serpent is a super-pseudorandom permutation.

1 Introduction

There are five AES finalists, RC6, Serpent, MARS, Twofish and Rijndael. RC6 was proposed by Rivest et.al. [6] as a successor of RC5. RC6 makes essential use of data-dependent rotations in the new structure. It also includes the use of four working registers and the inclusion of integer multiplication as an additional primitive operation. Serpent was proposed by Anderson et.al. [1]. Each round of Serpent has 32 parallel S-boxes and a following linear transformation of 128 bits. MARS was proposed by Burwick et.al. [2]. It uses a so called type-3 Feistel structure. Twofish was proposed by Schneier et.al. [7]. It has a 16 round Feistel structure. Rijndael was proposed by Daemen et.al. [3]. Its round transformation consists of three distinct invertible uniform transformations.

We consider the security of block ciphers in two ways, pseudorandomness and super-pseudorandomness.

- Pseudorandomness means that no attacker with polynomially many *encryption* queries can distinguish between the block cipher and a truly random permutation. This security corresponds to a chosen *plaintext* attack.
- Super-pseudorandomness means that no attacker with polynomially many *encryption and decryption* queries can distinguish between the block cipher and a truly random permutation. This security corresponds to a chosen *plaintext and ciphertext* attack.

B. Schneier (Ed.): FSE 2000, LNCS 1978, pp. 231–243, 2001.

Note that super-pseudorandomness implies pseudorandomness.

Luby and Rackoff idealized DES by replacing each *round function* with *one large* random function. Then they showed that the idealized three round DES yields a pseudorandom permutation and the idealized four round DES yields a super-pseudorandom permutation [4]. Maurer gave a simpler proof for non-adaptive adversaries [5].

For this kind of idealization, the three round idealized Twofish is a pseudorandom permutation and the four round idealized Twofish is a super-pseudorandom permutation because Twofish has the same Feistel structure as DES. MARS has a so called type-3 Feistel structure. At the rump session of AES2, Vaudenay and Moriai claimed that the five round idealized MARS is a pseudorandom permutation [8].

In this paper, we introduce a primitive-wise idealization in which some of the *primitive operations* of the round function (e.g., linear transformations and etc.) are left untouched and some of them (e.g., S-boxes and etc.) are replaced with *small* random functions or permutations. It is not known whether such a primitive-wise idealized DES is pseudorandom (or super-pseudorandom). Similarly, the same problem is open for all the AES candidates.

We solve this problem for RC6 partially, and solve for Serpent. We first idealize RC6 by replacing only an "$x \times (2x+1)$"operation with a pseudorandom function. The data-dependent rotation parts and the connections among the four registers are left untouched because they are the main properties of RC6. We then prove that the four round primitive-wise idealized RC6 is not a pseudorandom permutation for non-adaptive adversaries.

Serpent is idealized similarly. The linear transformation parts are left untouched and only the S-boxes are replaced with small pseudorandom permutations. We then prove that the two round primitive-wise idealized Serpent is not a pseudorandom permutation and the three round primitive-wise idealized Serpent is a super-pseudorandom permutation for non-adaptive adversaries.

A similar analysis for Rijndael, MARS, and Twofish is now in progress. Our results are stronger than the previous results for DES, Twofish [4] and MARS [8] because our idealization assumes weaker and smaller modifications of the ciphers.

This paper is organized as follows. In Section 2, we review the security model and the pseudorandomness of Twofish and MARS. The primitive-wise idealized RC6 is studied in Section 3 and the primitive-wise idealized Serpent is studied in Section 4.

2 Preliminaries

2.1 Security Model

Let us consider a computationally unbounded distinguisher $\mathcal{A}$ with an oracle $\mathcal{O}$. The oracle $\mathcal{O}$ chooses a permutation π randomly from the set of all permutations C^* over $\{0,1\}^n$ or from a subset of permutations $C \subset C^*$ (For a block cipher, C is the set of permutations obtained from all the keys). The aim of the distinguisher $\mathcal{A}$ is to distinguish if the oracle $\mathcal{O}$ implements C^* or C. Let p_{C^*} denote the probability that $\mathcal{A}$ outputs 1 when $\mathcal{O}$ implements C^* and p_C denote the probability that $\mathcal{A}$ outputs 1 when $\mathcal{O}$ implements C. That is,

$$p_{C^*} \stackrel{\triangle}{=} \Pr(\mathcal{A} \text{ outputs } 1 \mid \mathcal{O} \leftarrow C^*) \text{ and } p_C \stackrel{\triangle}{=} \Pr(\mathcal{A} \text{ outputs } 1 \mid \mathcal{O} \leftarrow C) \ .$$

Then the advantage $\texttt{Adv}_{\mathcal{A}}$ of the distinguisher $\mathcal{A}$ is defined as

$$\texttt{Adv}_{\mathcal{A}} \stackrel{\triangle}{=} |p_C - p_{C^*}| \ .$$

Suppose that $\mathcal{A}$ is limited to make at most $poly(n)$ queries to $\mathcal{O}$, where $poly(n)$ is some polynomial in n. We say that $\mathcal{A}$ is a pseudorandom distinguisher if it queries x and the oracle answers $y = \pi(x)$, where π is a randomly chosen permutation by $\mathcal{O}$. We say that $\mathcal{A}$ is a super-pseudorandom distinguisher if it is also allowed to query y and receives $x = \pi^{-1}(y)$ from the oracle.

Finally, C is called a pseudorandom permutation ensemble if $\texttt{Adv}_{\mathcal{A}}$ is negligible for any pseudorandom distinguisher (A pseudorandom function ensemble is defined similarly). C is called a super-pseudorandom permutation ensemble if $\texttt{Adv}_{\mathcal{A}}$ is negligible for any super-pseudorandom distinguisher. On the other hand, C^* is called the truly random permutation ensemble.

In this paper, we consider a non-adaptive distinguisher, i.e., a distinguisher that sends all the queries to the oracle at the same time.

2.2 Pseudorandomness of Idealized Twofish

Twofish has the same Feistel structure as DES shown in Fig. 1.

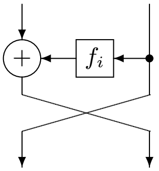

Fig. 1. The i-th round of the idealized Twofish

Assume that each round functions f_i is an independent pseudorandom function from $\{0,1\}^{n/2}$ to $\{0,1\}^{n/2}$. Then the following propositions are derived from the result of Luby and Rackoff [4].

Proposition 1. *The four round idealized Twofish is a super-pseudorandom permutation.*

Proposition 2. *The three round idealized Twofish is a pseudorandom permutation.*

2.3 Pseudorandomness of Idealized MARS

MARS has a structure as shown in Fig. 2.

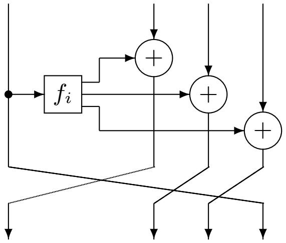

Fig. 2. The i-th round of the idealized MARS

Assume that each round function f_i is an independent pseudorandom function from $\{0,1\}^{n/4}$ to $\{0,1\}^{3n/4}$. Then Vaudenay and Moriai claimed the following proposition [8].

Proposition 3. *The five round idealized MARS is a pseudorandom permutation.*

3 Pseudorandomness of Primitive-Wise Idealized RC6

3.1 Primitive-Wise Idealization of RC6

RC6 is specified as RC6-$w/r/b$, where w denotes the number of bits of a word, r denotes the number of rounds, and b denotes the length of the encryption key in bytes. RC6 works with four w bits registers, A, B, C, and D. The i-th round of RC6 is defined as follows.

$$\begin{aligned}
&t = (B \times (2B+1)) \lll \lg w \\
&u = (D \times (2D+1)) \lll \lg w \\
&A = ((A \oplus t) \lll u) + S[2i] \\
&C = ((C \oplus u) \lll t) + S[2i+1] \\
&(A, B, C, D) = (B, C, D, A)
\end{aligned}$$

Definition of the i-th round of RC6

In the above definition, $a+b$ is an addition modulo 2^w, $a \oplus b$ is a bitwise exclusive-or of two w bits words, $a \times b$ is a multiplication modulo 2^w and $a \lll b$ denotes to rotate a w bits word a to the left by x, where x is the number given by the least significant lg w bits of b and lg w denotes the base-two logarithm of w. Finally, $S[2i]$ and $S[2i+1]$ denote the i-th round key.

Let n denote the length of a plaintext. Then $n = 4w$. In other words, each of A, B, C, D takes an element of $\{0,1\}^{n/4}$.

Now we idealize RC6 as shown below, where each f_j is an independent pseudorandom function from $\{0,1\}^{n/4}$ to $\{0,1\}^{n/4}$.

$$\begin{aligned}
&t = f_{2i}(B)\\
&u = f_{2i+1}(D)\\
&A = ((A \oplus t) \lll u)\\
&C = ((C \oplus u) \lll t)\\
&(A, B, C, D) = (B, C, D, A)
\end{aligned}$$

The i-th round of the primitive-wise idealized RC6

Note that

1. We replace t and $S[2i]$ with f_{2i}, and u and $S[2i+1]$ with f_{2i+1}.
2. However, we leave the data-dependent rotations $\lll t$, $\lll u$ and the connections among the four registers untouched because they are the main properties of RC6.

3.2 Pseudorandomness of Primitive-Wise Idealized RC6

The primitive-wise idealized RC6 is illustrated in Fig. 3, where $x = (x_0, x_1, x_2, x_3)$ denotes a plaintext, $z = (z_0, z_1, z_2, z_3)$ and $w = (w_0, w_1, w_2, w_3)$ denote ciphertexts of the three and four round primitive-wise idealized RC6, respectively. Each of x_i, z_i, and w_i is $n/4$ bits long.

Theorem 1. *The four round primitive-wise idealized RC6 is not a pseudorandom permutation.*

Proof. Let C be the set of permutations over $\{0,1\}^n$ obtained from the four round primitive-wise idealized RC6. We consider a distinguisher $\mathcal{A}$ such as follows.

1. $\mathcal{A}$ randomly chooses two plaintexts $x^{(1)} = (x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)})$ and $x^{(2)} = (x_0^{(2)}, x_1^{(2)}, x_2^{(2)}, x_3^{(2)})$ such that
$$x_0^{(1)} \neq x_0^{(2)} \text{ and } x_1^{(1)} = x_1^{(2)}, x_2^{(1)} = x_2^{(2)}, x_3^{(1)} = x_3^{(2)} \quad . \tag{1}$$
2. $\mathcal{A}$ sends them to the oracle and receives the ciphertexts $w^{(1)} = (w_0^{(1)}, w_1^{(1)}, w_2^{(1)}, w_3^{(1)})$ and $w^{(2)} = (w_0^{(2)}, w_1^{(2)}, w_2^{(2)}, w_3^{(2)})$ from the oracle.
3. Finally, $\mathcal{A}$ outputs 1 if and only if
$$((w_0^{(1)} \oplus w_0^{(2)}) \lll l) = x_0^{(1)} \oplus x_0^{(2)} \tag{2}$$
for some $0 \leq l < n/4$.

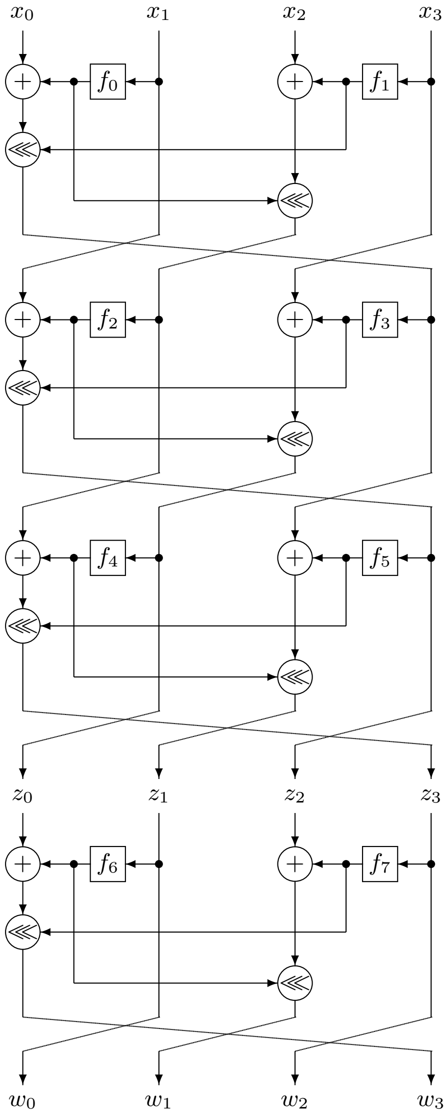

Fig. 3. The primitive-wise idealized RC6

Suppose that the oracle implements the truly random permutation ensemble C^*. Then for any fixed $x^{(1)}$ and $x^{(2)}$ satisfying (1),

$$\Pr(\mathcal{A} \text{ outputs } 1) = \frac{\#\{(w_0^{(1)}, w_0^{(2)}) \mid \text{eq.(2) holds for some } l\}}{\#\{(w_0^{(1)}, w_0^{(2)})\}} \ .$$

It is clear that

$$\#\{(w_0^{(1)}, w_0^{(2)})\} = (2^{n/4})^2 = 2^{n/2} \ .$$

For each $w_0^{(1)}$ and l, there exists a unique $w_0^{(2)}$ which satisfies eq.(2). Therefore,

$$\#\{(w_0^{(1)}, w_0^{(2)}) \mid \text{eq.(2) holds for some } l\} \le \frac{n}{4} \times 2^{n/4} \ .$$

Hence,

$$\Pr(\mathcal{A} \text{ outputs } 1) \le \frac{n/4 \times 2^{n/4}}{2^{n/2}} = \frac{n}{4 \times 2^{n/4}} \ .$$

Consequently,

$$p_{C^*} = E_{x^{(1)},x^{(2)}}(\Pr(\mathcal{A} \text{ outputs } 1)) \le \frac{n}{4 \times 2^{n/4}} \ .$$

Next suppose that the oracle implements the four round primitive-wise idealized RC6. We first assume that each f_i is a truly random function. Define α_1, β_1, δ_1 and γ_1 as shown in Fig. 4,

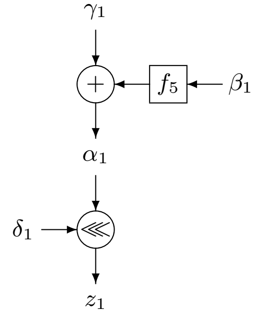

Fig. 4. 3-rd branch in the 3-rd round

Fix $x^{(1)} = (x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, x_3^{(1)})$ and $x^{(2)} = (x_0^{(2)}, x_1^{(2)}, x_2^{(2)}, x_3^{(2)})$ such that $x_0^{(1)} \neq x_0^{(2)}$, $x_1^{(1)} = x_1^{(2)}$, $x_2^{(1)} = x_2^{(2)}$ and $x_3^{(1)} = x_3^{(2)}$ arbitrarily. Then

$$\gamma_1^{(1)} \oplus \gamma_1^{(2)} = ((x_0^{(1)} \oplus x_0^{(2)}) \lll l')$$

for some l' since $x_1^{(1)} = x_1^{(2)}$, $x_2^{(1)} = x_2^{(2)}$ and $x_3^{(1)} = x_3^{(2)}$. If $\beta_1^{(1)} = \beta_1^{(2)}$, then $\alpha_1^{(1)} \oplus \alpha_1^{(2)} = \gamma_1^{(1)} \oplus \gamma_1^{(2)}$. Thus,

$$\alpha_1^{(1)} \oplus \alpha_1^{(2)} = ((x_0^{(1)} \oplus x_0^{(2)}) \lll l') \ .$$

Further if $\delta_1^{(1)} = \delta_1^{(2)}$, then

$$z_1^{(1)} \oplus z_1^{(2)} = ((\alpha_1^{(1)} \oplus \alpha_1^{(2)}) \lll l'')$$

for some l''. Therefore,

$$z_1^{(1)} \oplus z_1^{(2)} = ((x_0^{(1)} \oplus x_0^{(2)}) \lll l''')$$

for some l'''. Hence, if both $\beta_1^{(1)} = \beta_1^{(2)}$ and $\delta_1^{(1)} = \delta_1^{(2)}$ occur, then

$$((w_0^{(1)} \oplus w_0^{(2)}) \lll l) = x_0^{(1)} \oplus x_0^{(2)}$$

holds for some l because $z_1^{(1)} = w_0^{(1)}$ and $z_1^{(2)} = w_0^{(2)}$. Therefore,

$$p_C \geq \Pr(\beta_1^{(1)} = \beta_1^{(2)} \text{ and } \delta_1^{(1)} = \delta_1^{(2)}) \ .$$

Since $0 \leq l < n/4$ and f_4 is a truly random function, it is easy to see that

$$\Pr(\delta_1^{(1)} = \delta_1^{(2)}) \geq \frac{1}{n/4} \ .$$

Since $x_1^{(1)} = x_1^{(2)}$ and the output of f_2 for $x^{(1)}$ is equal to that for $x^{(2)}$,

$$\Pr(\beta_1^{(1)} = \beta_1^{(2)}) \geq \frac{1}{n/4} \ .$$

Further, β_1 and δ_1 are independent because f_4 is a truly random function. Consequently,

$$\begin{aligned} p_C &\geq \Pr(\beta_1^{(1)} = \beta_1^{(2)}) \times \Pr(\delta_1^{(1)} = \delta_1^{(2)}) \\ &\geq \frac{1}{n/4} \times \frac{1}{n/4} \\ &= \frac{16}{n^2} \ . \end{aligned}$$

Therefore, we obtain that

$$\mathtt{Adv}_{\mathcal{A}} = |p_C - p_{C^*}| \geq \frac{16}{n^2} - \frac{n}{4 \times 2^{n/4}} \ ,$$

which is non-negligible. Finally, we can show that $\mathtt{Adv}_{\mathcal{A}}$ is non-negligible even if each f_i is a pseudorandom function. The proof is almost the same as the proof of [4, Theorem 1]. Hence, the four round primitive-wise idealized RC6 is not a pseudorandom permutation. □

The above theorem implies that the four round primitive-wise idealized RC6 is not a super-pseudorandom permutation.

4 Pseudorandomness of Primitive-Wise Idealized Serpent

4.1 Primitive-Wise Idealization of Serpent

Serpent consists of 32 rounds. The plaintext becomes the first intermediate data B_0, after which the 32 rounds are applied, where each round i consists of three operations:

1. Key Mixing: At each round, a 128 bits subkey K_i is exclusive or'ed with the current intermediate data B_i.
2. S-Boxes: The 128 bits combination of input and key is considered as four 32 bits words. The S-box is applied to these four words, and the result is four output words. The CPU is employed to execute the 32 copies of the S-box simultaneously, resulting with $\mathcal{S}_i(B_i, K_i)$. Each S-box is a permutation over $\{0,1\}^4$.
3. Linear Transformation: The 32-bit in each of the output words are linearly mixed, by

$$\begin{aligned}
X_0, X_1, X_2, X_3 &:= \mathcal{S}_i(B_i, K_i) \\
X_0 &:= X_0 \lll 13 \\
X_2 &:= X_2 \lll 3 \\
X_1 &:= X_1 \oplus X_0 \oplus X_2 \\
X_3 &:= X_3 \oplus X_2 \oplus (X_0 \ll 3) \\
X_1 &:= X_1 \lll 1 \\
X_3 &:= X_3 \lll 7 \\
X_0 &:= X_0 \oplus X_1 \oplus X_3 \\
X_2 &:= X_2 \oplus X_3 \oplus (X_1 \ll 7) \\
X_0 &:= X_0 \lll 5 \\
X_2 &:= X_2 \lll 22 \\
B_{i+1} &:= X_0, X_1, X_2, X_3 \ ,
\end{aligned}$$

where $\lll$ denotes rotation, and $\ll$ denotes shift.

The effect of the linear transformation is that each plaintext bit affects all the data bits after three rounds. This can be detailed as follows. 4 output bits of some S-box in the first round are expanded by the linear transformation, so that they are input bits to m S-boxes in the second round. Then the $4m$ output bits of these m S-boxes are expanded so that they become input bits to the 32 S-boxes in the third round. The maximum value of m is 19, and the minimum is 17.

We idealize Serpent as shown in Fig. 5 and:

1. Let $n = 128 \times k$ denote the length of a plaintext.

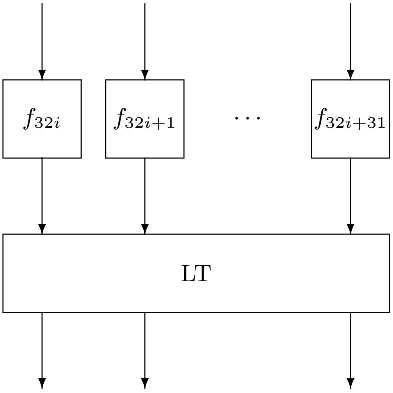

Fig. 5. The i-th round of the primitive-wise idealized Serpent

2. We assume that each f_i is an independent pseudorandom permutation over $\{0,1\}^{4k}$.
3. In the linear transformation, $a \lll b$ is replaced with $a \lll bk$, and $a \ll b$ is replaced with $a \ll bk$.

Note that we leave the linear transformation part untouched except the above modification.

4.2 Pseudorandomness of Primitive-Wise Idealized Serpent

The three round primitive-wise idealized Serpent is illustrated in Fig. 6. Let $x = (x_0, \ldots, x_{31})$ denote a plaintext, $z = (z_0, \ldots, z_{31})$ and $y = (y_0, \ldots, y_{31})$ denote ciphertexts of the two round and the three round primitive-wise idealized Serpent, respectively. Each of x_i, z_i, and y_i is $4k$ bits long.

We first prove the following theorem.

Theorem 2. *The two round primitive-wise idealized Serpent is not a pseudorandom permutation.*

Proof. Let C be the set of permutations over $\{0,1\}^n$ obtained from the two round primitive-wise idealized Serpent. We consider a distinguisher $\mathcal{A}$ such as follows.

1. $\mathcal{A}$ chooses two plaintexts, $x^{(1)} = (x_0^{(1)}, \ldots, x_{31}^{(1)})$ and $x^{(2)} = (x_0^{(2)}, \ldots, x_{31}^{(2)})$ such that $x_0^{(1)} \neq x_0^{(2)}$ and $x_1^{(1)} = x_1^{(2)}, \ldots, x_{31}^{(1)} = x_{31}^{(2)}$.
2. $\mathcal{A}$ sends them to the oracle and receives the ciphertexts $z^{(1)} = (z_0^{(1)}, \ldots, z_{31}^{(1)})$ and $z^{(2)} = (z_0^{(2)}, \ldots, z_{31}^{(2)})$ from the oracle.
3. $\mathcal{A}$ computes $v^{(1)} = \mathrm{LT}^{-1}(z^{(1)})$ and $v^{(2)} = \mathrm{LT}^{-1}(z^{(2)})$.

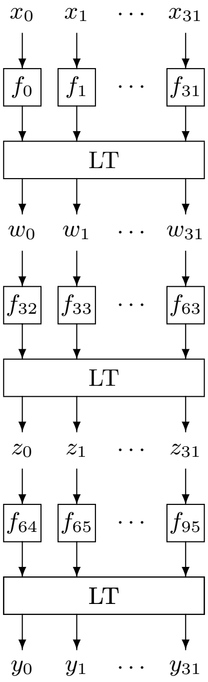

Fig. 6. The primitive-wise idealized Serpent

4. $\mathcal{A}$ outputs 1 if and only if $v_1^{(1)} = v_1^{(2)}$.

Suppose that the oracle implements the truly random permutation ensemble C^*. Then it is clear that $p_{C^*} = 1/2^{4k}$.

Next suppose that the oracle implements the two round primitive-wise idealized Serpent. The input to f_{33} includes no output of f_0. Therefore, $v_1^{(1)} = v_1^{(2)}$ because $x_1^{(1)} = x_1^{(2)}, \ldots, x_{31}^{(1)} = x_{31}^{(2)}$. Hence $p_C = 1$.

Therefore

$$\texttt{Adv}_{\mathcal{A}} = |p_C - p_{C^*}| = 1 - \frac{1}{2^{4k}} \enspace .$$

Consequently, $\texttt{Adv}_{\mathcal{A}}$ is non-negligible. Hence, the two round primitive-wise idealized Serpent is not a pseudorandom permutation. □

The above theorem implies that the two round primitive-wise idealized Serpent is not a super-pseudorandom permutation.

We next prove the following theorem.

Theorem 3. *The three round primitive-wise idealized Serpent is a pseudorandom permutation for non-adaptive adversaries.*

Proof. Let C be the set of permutations over $\{0,1\}^n$ obtained from the three round primitive-wise idealized Serpent. First, assume that each f_i is a truly random permutation.

Suppose that $\mathcal{A}$ makes p oracle calls. In the i-th oracle call, $\mathcal{A}$ sends a plaintext $x^{(i)} = (x_0^{(i)}, \ldots, x_{31}^{(i)})$ to the oracle and receives the ciphertext $y^{(i)} = (y_0^{(i)}, \ldots, y_{31}^{(i)})$ from the oracle. In Fig. 6, let $w^{(i)} = (w_0^{(i)}, \ldots, w_{31}^{(i)})$ denote the inputs to $f_{32}, \ldots, f_{63}$ and $z^{(i)} = (z_0^{(i)}, \ldots, z_{31}^{(i)})$ denote the inputs to $f_{64}, \ldots, f_{95}$.

Without loss of generality, we can assume that $x^{(1)}, \ldots, x^{(p)}$ are all distinct. Let $\mathcal{E}_{z_t}$ be the event that $z_t^{(1)}, \ldots, z_t^{(p)}$ are all distinct for $t = 0, \ldots, 31$, and let $\mathcal{E}_z$ be the event that all $\mathcal{E}_{z_0}, \ldots, \mathcal{E}_{z_{31}}$ occur. If $\mathcal{E}_z$ occurs, then, $y^{(1)}, \ldots, y^{(p)}$ are completely random since $f_{64}, \ldots, f_{95}$ are truly random permutations. Therefore, $\mathtt{Adv}_{\mathcal{A}}$ is upper bounded by

$$\mathtt{Adv}_{\mathcal{A}} = |p_C - p_{C^*}| \leq 1 - \Pr(\mathcal{E}_z) \ .$$

Further, it is easy to see that

$$1 - \Pr(\mathcal{E}_z) \leq \sum_{1 \leq i < j \leq p} \Pr(z_0^{(i)} = z_0^{(j)}) + \cdots + \sum_{1 \leq i < j \leq p} \Pr(z_{31}^{(i)} = z_{31}^{(j)}) \ . \qquad (3)$$

Fix $i \neq j$ arbitrarily. We show that all $\Pr(z_0^{(i)} = z_0^{(j)}), \ldots, \Pr(z_{31}^{(i)} = z_{31}^{(j)})$ are sufficiently small. Since $x^{(i)} \neq x^{(j)}$, we have $x_s^{(i)} \neq x_s^{(j)}$ for some $0 \leq s \leq 31$. For this s, f_s has $4k$ output bits. From the property of LT, the output bits of f_s are distributed among m w_t's, say $t = t_0, \ldots, t_{m-1}$, where m depends of s. Each w_t contains at least k bits of those from our modification of LT. Therefore,

$$\Pr(w_t^{(i)} = w_t^{(j)}) \leq \frac{1}{2^k}$$

for $t = t_0, \ldots, t_{m-1}$ because f_s is a truly random permutation.

Next each w_t becomes the input to f_{32+t}. The output bits of $f_{32+t_0}, \ldots, f_{32+t_{m-1}}$ are distributed among all of $z_0, \ldots, z_{31}$ from the property of LT. Each z_u contains at least k bits of those from our modification of LT.

Let $\mathcal{E}_w$ be the event that $w_t^{(i)} \neq w_t^{(j)}$ for $t = t_0, \ldots, t_{m-1}$. Then we have

$$\begin{aligned}\Pr(z_u^{(i)} = z_u^{(j)}) &\leq \frac{1}{2^k}\Pr(\mathcal{E}_w) + (1 - \Pr(\mathcal{E}_w)) \\ &\leq \frac{1}{2^k} + \Pr(w_{t_0}^{(i)} = w_{t_0}^{(j)}) + \cdots + \Pr(w_{t_{m-1}}^{(i)} = w_{t_{m-1}}^{(j)}) \\ &\leq \frac{1}{2^k} + \frac{m}{2^k}\end{aligned}$$

for $u = 0, \ldots, 31$. Therefore, the right side of (3) is upper bounded as follows.

$$\begin{aligned}\sum_{1 \leq i < j \leq p} \Pr(z_0^{(i)} = z_0^{(j)}) + \cdots + \sum_{1 \leq i < j \leq p} \Pr(z_{31}^{(i)} = z_{31}^{(j)}) &\leq \frac{16(m+1)p^2}{2^k} \\ &\leq \frac{320 \times p^2}{2^{n/128}} \ ,\end{aligned}$$

because $m \leq 19$ and $n = 128 \times k$.

Since $p = poly(n)$, $\mathtt{Adv}_{\mathcal{A}}$ is negligible for any $\mathcal{A}$. Finally, we can show that $\mathtt{Adv}_{\mathcal{A}}$ is negligible even if each f_i is a pseudorandom permutation as the proof of [4, Theorem 1]. □

We can prove the following corollary similarly.

Corollary 1. *The three round primitive-wise idealized Serpent is a super-pseudorandom permutation for non-adaptive adversaries.*

References

1. R.Anderson, E.Biham and L.Knudsen. *Serpent: a proposal for the Advanced Encryption Standard*. AES proposal, available on: `http://www.cl.cam.ac.uk/~rja14/serpent.html`.
2. C.Burwick, D.Coppersmith, E.D'Avignon, R.Gennaro, S.Halevi, C.Jutla, S.M.Matyas Jr., L.O'Connor, M.Peyravian, D.Safford and N.Zunic. *MARS — a candidate cipher for AES*. AES proposal, available on: `http://www.research.ibm.com/security/mars.html`.
3. J.Daemen and V.Rijmen. *AES proposal: Rijndael*. AES proposal, available on: `http://www.esat.kuleuven.ac.be/~rijmen/rijndael/`.
4. M.Luby and C.Rackoff. *How to construct pseudorandom permutations from pseudorandom functions*. SIAM Journal on Computing, volume 17, number 2, pages 373–386, April 1988.
5. U.M.Maurer. *A simplified and generalized treatment of Luby-Rackoff pseudorandom permutation generators*. Advances in Cryptology — Eurocrypt '92, Lecture Notes in Computer Science, volume 658, pages 239–255, Springer-Verlag, 1992.
6. R.L.Rivest, M.J.B.Robshaw, R.Sidney and Y.L.Yin. *The RC6 Block Cipher. v1.1*. AES proposal, available on: `http://www.rsa.com/rsalabs/aes/`.
7. B.Schneier, J.Kelsey, D.Whiting, D.Wagner, C.Hall and N.Ferguson. *Twofish: a 128-bit block cipher*. AES proposal, available on: `http://www.counterpane.com/twofish.html`.
8. S.Vaudenay and S.Moriai. *Comparison of the randomness provided by some AES candidates*. Rump session at AES2.