IEICE TRANS. INF. & SYST., VOL.E96–D, NO.1 JANUARY 2013

LETTER

Round Addition Using Faults for Generalized Feistel Network

Hideki YOSHIKAWA[†a)], **Masahiro KAMINAGA**[†], ***and*** **Arimitsu SHIKODA**[†], ***Members***

SUMMARY This article presents a differential fault analysis (DFA) technique using round addition for a generalized Feistel network (GFN) including CLEFIA and RC6. Here the term "round addition" means that the round operation executes twice using the same round key. The proposed DFA needs bypassing of an operation to count the number of rounds such as increment or decrement. To verify the feasibility of our proposal, we implement several operations, including increment and decrement, on a microcontroller and experimentally confirm the operation bypassing. The proposed round addition technique works effectively for the generalized Feistel network with a partial whitening operation after the last round. In the case of a 128-bit CLEFIA, we show a procedure to reconstruct the round keys or a secret key using one correct ciphertext and two faulty ciphertexts. Our DFA also works for DES and RC6.

key words: *differential fault analysis (DFA), round addition, block cipher, generalized Feistel network (GFN), DES, CLEFIA, RC6*

1. Introduction

Differential fault analysis (DFA) is an effective attack technique against symmetric and asymmetric encryption algorithms. Various DFA techniques and countermeasures to implement ciphers using smartcard-like devices have been developed [1]–[5]. Choukri and Tunstall [2] reported successful "round reduction" by bypassing a branch operation to check the number of rounds of AES implemented on a PIC16F877 microcontroller produced by Microchip. They used a glitch on the power supplied to the microcontroller as their fault injection method. Moreover, Park et al. [3] obtained similar results using laser a beam injection with an ATmega128 microcontroller produced by Atmel.

On the other hand, Kaminaga et al. [6] show the effectiveness of a combination of multiple branches and a brown-out detector (BOD), which is a type of low voltage detector, in their experiment using the ATmega168 (Atmel, USA) microcontroller. According to their experiment, round reduction did not work against the combination of triple branches and BOD. In this case, the execution time of multiple branches must be longer than the idle time of BOD. However, generally, an increment or decrement operation is located before or after a conditional branch operation, and these operations cannot be multiplied directly. If an increment or decrement operation is bypassed, the implemented cipher executes the round operation twice using the same round key, i.e., "round addition" must have occurred.

Manuscript received March 5, 2012.
Manuscript revised September 16, 2012.

[†]The authors are with the Faculty of Engineering, Tohoku Gakuin University, Tagajo-shi, 985–8537 Japan.
a) E-mail: hyoshi@tjcc.tohoku-gakuin.ac.jp

DOI: 10.1587/transinf.E96.D.146

In this article, we show the successful experimental results of operation bypassing shown in Fig. 3 by instantaneously supplying low voltage power. Based on these results, we present round keys or a secret key-extracting technique using round addition for Feistel or generalized Feistel network structured block ciphers including DES, CLEFIA, and RC6.

2. Round Addition Model

2.1 Encryption Algorithm for a Block Cipher System

Figure 1 shows the entire encryption process of a GFN structured block cipher. Each round consists of an F-function and a swap function, μ. The encryption process consists of pre-whitening (PrW), r-times round operations, and post-whitening (PoW). The cipher operation can be expressed as follows.

$$C = PoW \circ \mu \circ F_{RK_r} \circ \cdots \circ \mu \circ F_{RK_1} \circ PrW(P),$$

where P is plaintext, C is ciphertext derived from P, RK_i ($i = 1, 2, \cdots, r$) are round keys and $\circ$ is the composition operator. As explained, "round addition" means executing a round operation of the cipher twice with same round key, i.e., the i-th round added faulty ciphertext $C^{(i)}$ is given by

$$C^{(i)} = PoW \circ \mu \cdots \underline{\mu \circ F_{RK_i}} \circ \mu \circ F_{RK_i} \cdots \circ F_{RK_1} \circ PeW(P)$$

Figure 2 shows the attack point in r-round cipher operations.

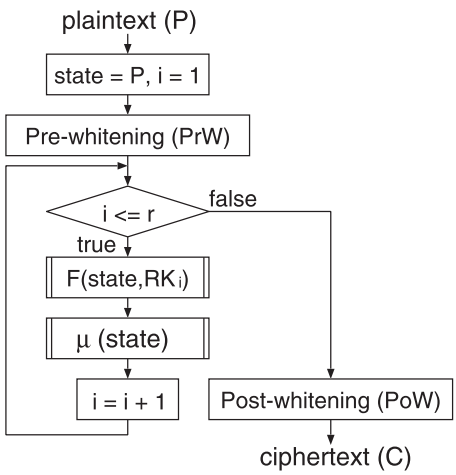

Fig. 1 Operation of a general block cipher.

Copyright © 2013 The Institute of Electronics, Information and Communication Engineers

```
state = PrW(P);
i = 1
while ( i <= r ) {
  F(state, RKi);
  μ (state);
  i = i + 1;            <--- Attack Point
}
C = PoW(state);
```

Fig. 2 Pseudocode for round-added encryption.

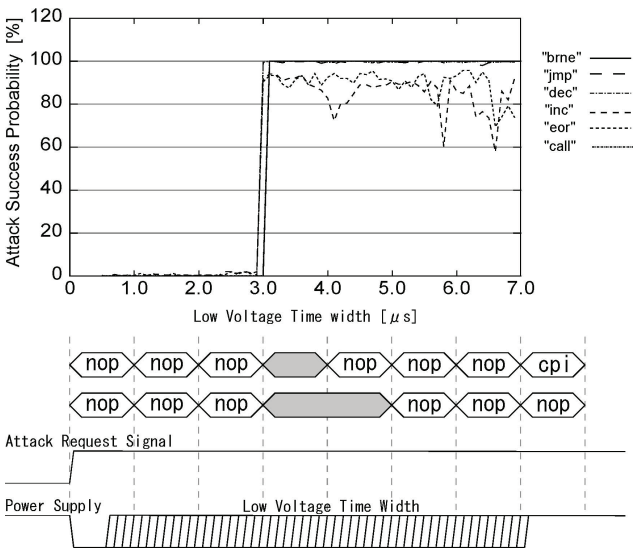

Fig. 3 Timing chart and success probabilities of command bypassing.

That is, round addition can be realized by bypassing the increment or decrement to count the number of rounds.

2.2 Experimental Results of Operation Bypassing

To verify the feasibility of our proposal, we implemented several operations including increment and decrement, on the ATmega168 (Atmel, USA) microcontroller and attempted to bypass these operations experimentally. Then, we succeeded bypassing these operations by supplying low voltage power instantaneously. Figure 3 shows the relation between attack timing and attack (bypass) success probabilities for branch not equal (brne), unconditional jump (jmp), decrement (dec), increment (inc), exclusive-or (eor), and call for subroutine (call). This experimental process is executed 400 times in 100 ns increments in order to determine the delay parameters and the low-voltage period needed to automate the tedious process of fault injection. In this figure, we can see that the attack success probabilities of "brne", "jmp", "dec", and "call" are almost 100%. The result shows that our attack using these command bypassing is feasible with a high probability. In contrast, the attack success probability of "inc" is more than 58 %, but it is lower than other commands. The increment command is the attack point in Fig. 2. Thus, we examine the attack feasibility of round addition using "inc" bypassing.

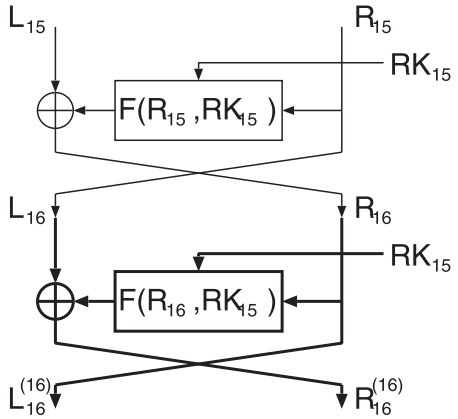

Fig. 4 Round addition attack at final round of DES.

3. Round Addition DFA

3.1 DES

In this subsection, we show that keys can be easily derived by a round addition attack on the data encryption standard (DES). The correct and faulty ciphertexts are denoted by $C = L_{16}|R_{16}$ and $C^{(16)} = L_{16}^{(16)}|R_{16}^{(16)}$, respectively. The concatenation is denoted by "|." Figure 4 illustrates the round addition attack at the final round operation. In this case, both ciphertexts C and $C^{(16)}$ are known. Thus, the final round key can be given by

$$F(R_{16}, RK_{15}) = L_{16} \oplus R_{16}^{(16)}$$

where $\oplus$ is denoted by bitwise exclusive-OR.

3.2 128 bit-CLEFIA

CLEFIA is a 128-bit block cipher developed by SONY Corporation in 2007. The advantages of CLEFIA are as follows: small implementation size and high speed utilizing its characteristic structure [7]. Here we show that the secret key of the 128-bit CLEFIA can be extracted by bypassing the double inc or dec operation. The following notations are used.

- $C = V|X|Y|Z$: 128-bit correct ciphertext constructed by the concatenation of four 32-bit data.
- V_i, X_i, Y_i, Z_i : Each i-th round output of the above correct ciphertext
- $C^{(17)} = V^{(17)}|X^{(17)}|Y^{(17)}|Z^{(17)}$: 17-th round added 128-bit faulty ciphertext
- $\dot{V}_i, \dot{X}_i, \dot{Y}_i, \dot{Z}_i$: Each i-th round output of the above faulty ciphertext
- $C^{(17^2)} = V^{(17^2)}|X^{(17^2)}|Y^{(17^2)}|Z^{(17^2)}$: 17-th double round added 128-bit faulty ciphertext
- $\ddot{V}_i, \ddot{Z}_i, \ddot{Y}_i, \ddot{Z}_i$: Each i-th round output of above faulty ciphertext

Figure 5 shows the encryption process of 128-bit CLEFIA. In this figure, the round keys are denoted by

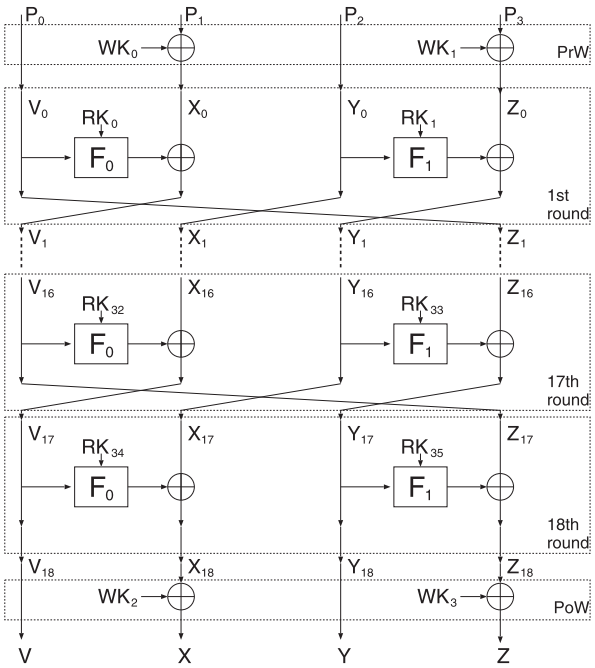

Fig. 5 Encryption process of 128 bit-CLEFIA.

```
state = PrW(P);
i = 1
while ( i <= 17 ) {
  F0(state, RK_{2i-2}); F1(state, RK_{2i-1});
  μ (state);
  i = i + 1;                    <--- Attack Point
}
F0(state, RK34); F1(state, RK35);
C = PoW(state);
```

Fig. 6 Pseudocode of 128-bit CLEFIA.

$RK_0, RK_1, \ldots, RK_{34}, RK_{35}$, and the whitening keys are denoted by WK_0, WK_1, WK_2, WK_3. The final round is different from another round, thus the pseudocodes can be described by Fig. 6.

We first obtain the following output by bypassing 17-th increment operation since the F-function executes twice. From Figs. 5 and 7, we can see following relations.

$$V = V_{18} = V_{17}, \quad Y = Y_{18} = Y_{17}$$
$$V^{(17)} = \dot{V}_{18} = \dot{V}_{17}, \quad \dot{X}_{17} = Y_{17}, \quad Y^{(17)} = \dot{Y}_{18} = \dot{Y}_{17},$$
$$\dot{Z}_{17} = V_{17}.$$

Next, we obtain the following output by bypassing twice at 17-th increment operation as depicted in Fig. 8.

$$V^{(17^2)} = \ddot{V}_{18} = \ddot{V}_{17}, \quad \ddot{X}_{17} = \dot{Y}_{17}, \quad Y^{(17^2)} = \ddot{Y}_{18} = \ddot{Y}_{17},$$
$$\ddot{Z}_{17} = \dot{V}_{17}.$$

Thus, the round keys, RK_{32} and RK_{33} can be extracted by the parts of correct and faulty ciphertexts as

$$RK_{32} = F_0^{-1}\left(Y \oplus V^{(17^2)}, V^{(17)}\right), \tag{1}$$

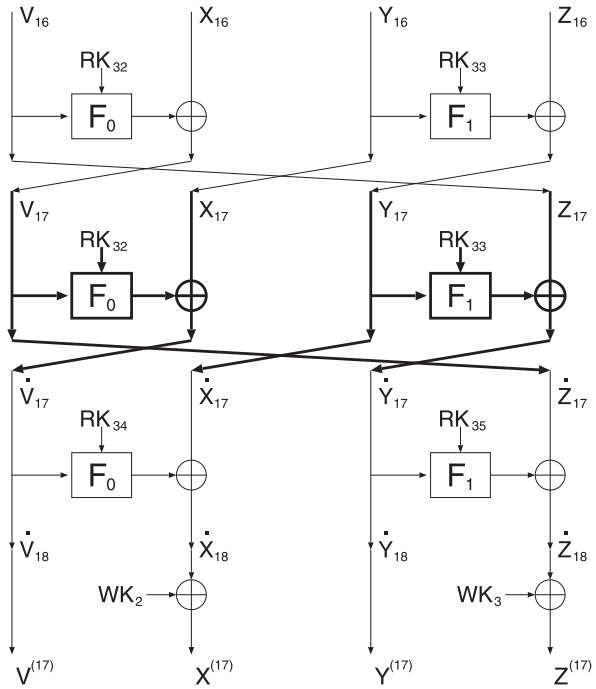

Fig. 7 Single round addition attack before the final round.

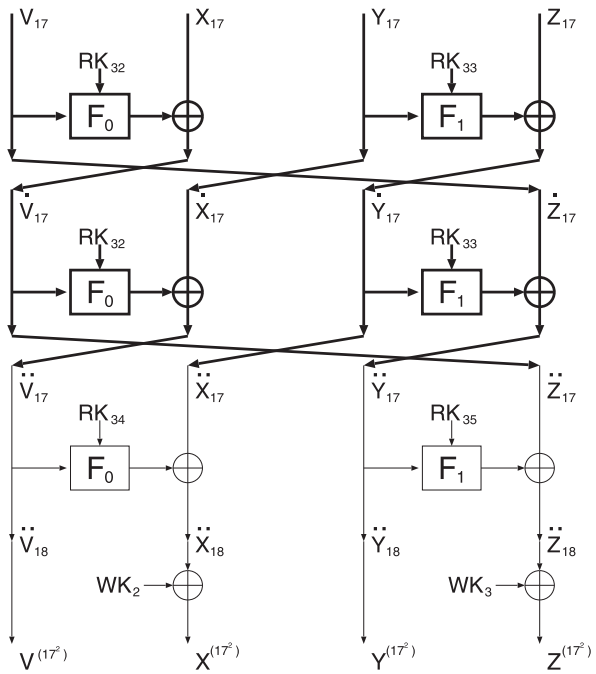

Fig. 8 Double round addition attack before the final round.

$$RK_{33} = F_1^{-1}\left(V \oplus Y^{(17^2)}, Y^{(17)}\right), \tag{2}$$

where $F_0^{-1}(\cdot,\cdot)$, $F_1^{-1}(\cdot,\cdot)$ are denoted by inverse function $F_0(\cdot,\cdot)$, $F_1(\cdot,\cdot)$, respectively. After determining RK_{32} and RK_{33}, the following relations are derived

$$X_{17} = V^{(17)} \oplus F_0(V, RK_{32}), \tag{3}$$
$$Z_{17} = Y^{(17)} \oplus F_1(Y, RK_{33}). \tag{4}$$

From Figs. 5 and 7, we can see

$$X = X_{18} \oplus WK_2 = X_{17} \oplus F_0(V, RK_{34}) \oplus WK_2, \quad (5)$$
$$X^{(17)} = \dot{X}_{18} \oplus WK_2 = \dot{X}_{17} \oplus F_0(V^{(17)}, RK_{34}) \oplus WK_2. \quad (6)$$

By adding Eqs. (5) (6) to eliminate the whitening key WK_2, we obtain

$$F_0(V, RK_{34}) \oplus F_0(V^{(17)}, RK_{34}) = X \oplus X^{(17)} \oplus X_{17} \oplus Y. \quad (7)$$

Using Eqs. (3) and (7), the round key RK_{34} can be solved easily by brute force. Similarly, using Eq. (4), RK_{35} can be derived by

$$F_1(Y, RK_{35}) \oplus F_1(Y^{(17)}, RK_{35}) = Z \oplus Z^{(17)} \oplus Z_{17} \oplus V. \quad (8)$$

After determining the round keys $RK_{32}, \ldots, RK_{35}$ by Eqs. (1), (2), (7), and (8), the secret key can be reconstructed by the key schedule procedure using the CLEFIA constants CON_i, $i = 0, \ldots 59$ [7].

3.3 RC6

The block cipher RC6 is an "advanced encryption standard (AES) finalist" [8]. Here, we show that the extended keys can be derived by each round addition attack. The 128-bit correct and the faulty ciphertexts by j-th operation added are denoted by $C = V|X|Y|Z$ and $C^{(j)} = V^{(j)}|X^{(j)}|Y^{(j)}|Z^{(j)}$, respectively. The correct and faulty operations are illustrated by Figs. 9 and 10, respectively. In these figures, the following relations are derived

$$S[2r+2] = V^{(r)} - X, \qquad S[2r+3] = Y^{(r)} - Z$$
$$S[2r] = Z^{(r)} -$$
$$\langle\langle\left(V - V^{(r)} + X + \langle\langle(f(X), 5), \langle\langle(f(Z), 5)\right)$$
$$S[2r+1] = X^{(r)} -$$
$$\langle\langle\left(Y - Y^{(r)} + Z + \langle\langle(f(Z), 5), \langle\langle(f(X), 5)\right)$$

where r is the number of rounds, $\langle\langle(a, b)$ denotes b-bit left rotation of the word a, and $f(t) = t(2t+1)$. Each extended keys $S[2], \ldots, S[2r+3]$ can be derived by increment operation bypassing each round operation. A pair of plain text and ciphertext is required to find the initial extended keys $S[0]$ and $S[1]$.

3.4 Experimental Result of CLEFIA

In this subsection, the experimental result of the command bypassing attack for the 128-bit CLEFIA implemented in an ATmega168 microcontroller is shown. When we set a 128-bit plaintext to be "0x00010203", "0x04050607", "0x08090A0B", "0x0C0D0E0F", with a 128-bit secret key K = "0xFFEEDDCC", "0xBBAA9988", "0x77665544", "0x33221100", the corresponding 128-bit correct ciphertext is V = "0xDE2BF2FD", X = "0x9B74AACD", Y = "0xF1298555", Z = "0x459494FD". These test vectors can be seen in [7]. Figure 7 and 8 show single and double round addition using "inc" command bypassing attacks respectively. We obtain a 128-bit faulty ciphertexts of $V^{(17)}$ = "0x02D7277D", $X^{(17)}$ = "0xF9329144", $Y^{(17)}$ = "0x5AE13C43", $Z^{(17)}$ = "0x4F83B00A" for single round addition, and $V^{(17^2)}$ = "0xF004463C", $X^{(17^2)}$ = "0x64574C22", $Y^{(17^2)}$ = "0x044D3EA7", $Z^{(17^2)}$ = "0x248FB0BE" for double round addition, respectively.

The successful command bypassing is defined by that a command bypassing occurs without any corruption of the intermediate data in the registers. When both the single and the double round addition are achieved by the successful by-

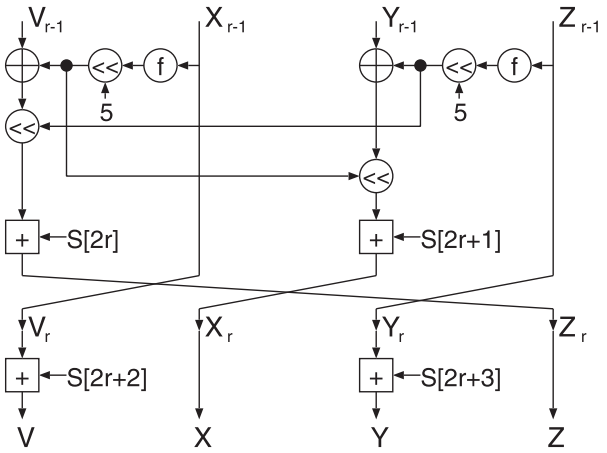

Fig. 9 Final round process of RC6.

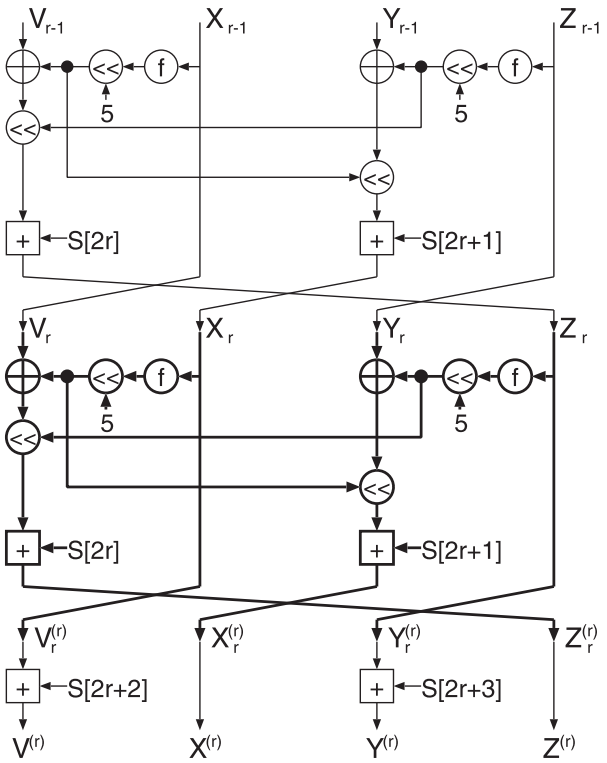

Fig. 10 Round addition attack at the final round of RC6.

passing of "inc" command just before the final round, we would obtain the round keys $RK_{32}, \ldots, RK_{35}$ using the procedure in Sect. 3.2. Since the single round addition is easier than the double round addition, the possibility of the key extraction mainly depends on the double round addition. From this point of view, the double "inc" command bypassing must be examined. Our results show that the double addition attack succeeded 60 times in 1000 trials. The experimental result was obtained in the same manner with [6]. In this result, we conclude that the round addition by command bypassing is feasible for practical block cipher systems.

4. Conclusion

In this article, we show the round keys or a secret key-extracting technique using round addition for the block cipher algorithm. Our round addition DFA works for DES, CLEFIA, and RC6. In particular, the results show that the block cipher algorithm should whiten all round data at the final round. For example, in CLEFIA, the first or third line are not whitened by adding round keys. Thus, we conclude that these structures have vulnerabilities as a result of operation bypassing.

References

[1] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," Proc. CRYPTO'97, LNCS 1294, pp.513–525, 1997.
[2] H. Choukri and M. Tunstall, "Round reduction using faults," Proc. FDTC, pp.13–24, 2005.
[3] J. Park, S. Moon, D. Choi, Y. Kang, and J. Ha, "Differential fault analysis for round-reduced AES by fault injection," ETRI J., vol.33, no.3, pp.434–441, June 2011.
[4] J. Takahashi and T. Fukunaga, "Differential fault analysis on CLEFIA with 128, 192, and 256-bit keys," IEICE Trans. Fundamentals, vol.E93-A, no.1, pp.136–143, Jan. 2010.
[5] R. Li, B. Sun, C. Lin, and J. You, "Differential fault analysis on SMS4 using single fault," Inf. Process. Lett., vol.111, pp.156–163, 2011.
[6] M. Kaminaga, A. Shikoda, and H. Yoshikawa, "Development and evaluation of a microstep DFA vulnerability estimation method," IEICE Electron. Express, vol.8, no.22, pp.1899–1904, Nov. 2011.
[7] Sony Corporation, "The 128-bit blockcipher CLEFIA algorithm specification," http://www.sony.net/Products/clefia/
[8] R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, "The RC6 block cipher (v1.1 Aug. 1998)," http://people.csail.mit.edu/rivest/Rc6.pdf